

IMPLEMENTASI PERBANDINGAN SISTEM PENGIRIMAN SINYAL SUARA SECARA WIRELESS PADA ARDUINO

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Muhammad Rasyid Perdana

NIM: 125150300111069



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019**

PENGESAHAN

IMPLEMENTASI PERBANDINGAN SISTEM PENGIRIMAN
SINYAL SUARA SECARA *WIRELESS* PADA ARDUINO

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Rasyid Perdana
NIM: 125150300111069

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Juli 2019
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Wijaya Kurniawan, S.T., M.T.
NIP: 19820125 201504 1 002

Rakhmadhany Primananda, S.T, M.Kom.
NIK: 201609 860406 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 6 Agustus 2019

Muhammad Rasyid Perdana

NIM: 125150300111069

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena Rahmat-Nya lah penulis dapat menyelesaikan skripsi ini. Adapun maksud penyusunan skripsi ini adalah untuk memenuhi salah satu persyaratan dalam menempuh ujian Sarjana Komputer pada Fakultas Ilmu Komputer. Judul skripsi yang disusun adalah: "Implementasi Perbandingan Sistem Pengiriman Sinyal Suara Secara Wireless Pada Arduino".

Banyak kesulitan dan hambatan yang dialami oleh penulis dalam menyusun skripsi ini terutama dalam meredupkan lampu dan mengolahnya, tetapi semua itu telah dapat diatasi dengan baik berkat dukungan dan bantuan dari berbagai pihak. Untuk itulah pada kesempatan ini penulis menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua orangtua dan seluruh keluarga besar atas segala nasehat, kasih sayang dan kesabaran dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
2. Bapak Wijaya Kurniawan, S.T., M.T. selaku dosen pembimbing I skripsi dan bapak Rakhmadhany Pramananda, S.T, M.Kom. selaku dosen pembimbing II skripsi yang dengan sabar membimbing, memberikan pengarahan dan memberi dukungan semangat sehingga dapat menyelesaikan skripsi ini.
3. Bapak Sabriansyah Rizqika Akbar, S.T., M.Eng. selaku ketua Program Studi Teknik Informatika. Universitas Brawijaya Malang.
4. Seluruh civitas akademika Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
5. Rekan – rekan Sistem Komputer yang selalu memberikan motivasi, dukungan serta bantuan sampai terselesaikannya skripsi ini.

Penulis menyadari bahwa proposal skripsi ini jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun penulis ucapkan terima kasih. Penulis mengharapkan semoga skripsi ini dapat bermanfaat bagi yang membutuhkan dan menggunakannya.

Malang, 8 juli 2019

Penulis

Ravehunterx785@gmail.com

ABSTRAK

Muhammad Rasyid Perdana, Implementasi Perbandingan Sistem Pengiriman Sinyal Suara Secara Wireless Pada Arduino

Pembimbing: Wijaya Kurniawan, S.T., M.T. dan Rakhmadhany Primananda, S.T, M.Kom.

Teknologi selalu mengalami perkembangan dari masa ke masa, dan berkembangnya teknologi pun mempengaruhi berbagai aspek dalam kehidupan, pada fokus kali ini bagaimana dampak teknologi terhadap media audio. Audio merupakan penyampai informasi yang berupa sinyal suara. dengan berkembangnya teknologi sekarang, tidak jarang audio digunakan sebagai media penyampaian pesan. Dalam mendapatkan kualitas suara yang baik perlu diperhatikan beberapa aspek aspek yang mempengaruhi kualitas suara seperti media penghubung dan penempatan sumber suara dan penerima suara. Pada beberapa kasus untuk pengiriman suara masih menggunakan kabel, jadi ada kalanya terdapat keterbatasan media penyampaian suara. Untuk itu penelitian ini merancang sebuah alat pengiriman suara secara wireless. Alat ini mempunyai transmitter sebagai pengiriman suara dan receiver sebagai penerimaan suara. Untuk pengiriman suara melalui media radio digunakan modul nRFL201, modul HC-02 untuk pengiriman suara melalui bluetooth, dan modul ESP8266 untuk pengiriman suara melalui wifi. Pembacaan modul akan diproses menggunakan mikro kontroler Arduino Uno. Pada tiap transmitter akan digunakan mic sebagai input suara dan untuk receiver digunakan speaker sebagai output suara. Berdasarkan Hasil pengujian menunjukkan perbandingan yang cukup besar saat mengeksekusi alat dan program secara bersama-sama. Hasil pengujian ini menunjukkan pada nrf setiap sinyal suara yang dikirimkan mempunyai data yang berbeda secara signifikan. Setiap perbedaan jarak maka data yang didapat mengalami penurunan. Sedangkan pada bluetooth sinyal suara yang dihasilkan relatif stabil. Tetapi terjadi penurunan kualitas suara. Untuk pengujian pada wifi hasil yang didapat pengiriman suara relatif stabil. Kualitas suara yang dihasilkan juga konsisten meskipun dilakukan pada jarak yang berbeda.

Kata kunci: sinyal suara, wireless, modul nRFL201, modul HC-02 ,modul ESP8266.

ABSTRACT

Muhammad Rasyid Perdana, Implementation Comparison of Arduino Wireless Signal Sending System

Adviser: Wijaya Kurniawan, S.T., M.T. dan Rakhmadhany Primananda, S.T, M.Kom.

Technology has always experienced development from time to time, and the development of technology also affects various aspects of life, at this time the focus of how the impact of technology on audio media. Audio is the transmitter of information in the form of sound signals. with the development of technology now, not infrequently audio is used as a medium for delivering messages. In getting good sound quality, it is necessary to pay attention to several aspects that affect sound quality such as connecting media and the placement of sound sources and voice receivers. For this reason, this study designed a wireless voice sending device. This tool has a transmitter as sending voice and receiver as voice reception. For sending voice via radio media, the nRFL201 module is used, the HC-02 module is for sending voice via bluetooth, and the ESP8266 module is for sending voice over wifi. The module reading will be processed using the Arduino Uno micro controller. In each transmitter the mic will be used as the voice input and the speaker will use the speaker as the sound output. Based on the test results it shows a fairly large comparison when executing tools and programs together. The results of this test indicate that each sound signal that is sent has significantly different data. Every difference in distance, the data obtained has decreased. Whereas in bluetooth the sound signal produced is relatively stable. But there was a decrease in sound quality. For testing on wifi the results obtained by voice delivery are relatively stable. The sound quality produced is also consistent even though it is done at different distances.

Keywords: *sound signal, wireless, modul nRFL201, modul HC-02 ,modul ESP8266.*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	2
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan	2
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	6
2.2.1 Sinyal Audio.....	6
2.2.3 Arduino Uno	7
2.2.4 ESP8266 WiFi Module	9
2.2.5 HC-05 Bluetooth Modul	10
2.2.6 nRF24L01.....	11
BAB 3 METODOLOGI	12
3.1 Alur Metodologi Penelitian.....	12
3.2 Studi Literatur	13
3.3 Rekayasa Kebutuhan Sistem	13
3.4 Perancangan Sistem.....	14
3.4.1 Perancangan perangkat keras.....	14

3.4.2 Perancangan perangkat lunak.....	15
3.5 Implementasi Sistem	15
3.6 Pengujian dan Analisis	15
3.6.1 Pengujian nRF24L01	15
3.6.2 Pengujian Bluetooth.....	15
3.6.3 Pengujian Wifi	15
3.6.5 Pengujian Keseluruhan sistem	15
3.7 Kesimpulan.....	16
BAB 4 REKAYASA KEBUTUHAN	17
4.1 Deskripsi Umum.....	17
4.1.1 Perspektif sistem	17
4.1.2 Kegunaan Sistem	17
4.1.3 Karakteristik pengguna	17
4.1.4 Lingkungan Operasi.....	17
4.1.5 Batasan Sistem	17
4.2 Rekayasa Kebutuhan Sistem	18
4.2.1 Kebutuhan perangkat keras	18
4.2.2 Kebutuhan perangkat lunak.....	18
4.2.3 Kebutuhan Fungsional.....	18
4.2.4 Kebutuhan Non Fungsional.....	19
4.3 Kebutuhan antarmuka	19
4.3.1 Kebutuhan Antarmuka Pengguna	19
4.3.2 Kebutuhan Antarmuka Perangkat Keras.....	19
4.3.3 Kebutuhan Antarmuka Perangkat Lunak	19
BAB 5 PERANCANGAN DAN IMPLEMENTASI	20
5.1 Perancangan Sistem.....	20
5.1.1 Gambaran Umum Sistem	20
5.1.2 Perancangan Perangkat Keras	20
5.1.3 Perancangan Perangkat Lunak.....	26
5.2 Implementasi	29
5.2.1 Implementasi Perangkat Keras	29
5.2.2 Implementasi Perangkat Lunak.....	32

BAB 6 PENGUJIAN DAN ANALISIS.....	37
6.1 Pengujian pada NRF24L01	37
5.2.2 Tujuan	37
5.2.2 Prosedur	38
5.2.2 hasil dan Analisa	40
6.2 Pengujian pada Bluetooth	40
5.2.2 Tujuan	40
5.2.2 Prosedur	40
5.2.2 hasil dan Analisa	42
6.3 Pengujian pada Wifi	43
5.2.2 Tujuan	43
5.2.2 Prosedur	43
5.2.2 hasil dan Analisa	45
6.4 Pengujian Keseluruhan	45
BAB 7 Penutup	48
7.1 Kesimpulan.....	48
7.2 Saran	48
DAFTAR PUSTAKA.....	49
LAMPIRAN	51

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian	5
Tabel 5.1 Hubungan Pinout nRF24L01 Pada Arduino Uno	21
Tabel 5.2 Hubungan Pinout HC-05 Pada Arduino Uno	22
Tabel 5.3 Hubungan Pinout ESP 8266 Pada Arduino Uno	23
Tabel 5.4 Hubungan Pinout nRF24L01 Pada Arduino Uno	24
Tabel 5.5 Hubungan Pinout HC-05 Pada Arduino Uno	25
Tabel 5.6 Hubungan Pinout ESP 8266 Pada MCP4725	26
Tabel 5.7 Program nrf	32
Tabel 5.8 Program transmit bluetooth	33
Tabel 5.9 Program reciever bluetooth	33
Tabel 5.10 Program komunikasi antarwifi	34
Tabel 5.11 Program transmit wifi	35
Tabel 5.12 Program reciever wifi	35
Tabel 6.1 Hasil Pengujian pada nrf.....	39
Tabel 6.2 Hasil Pengujian pada Bluetooth	42
Tabel 6.3 Hasil Pengujian pada Wifi.....	45
Tabel 6.4 Perbandingan Pengujian modul dengan jarak 1 meter	46
Tabel 6.5 Perbandingan Pengujian modul dengan jarak 3 meter	46
Tabel 6.6 Perbandingan Pengujian modul dengan jarak 6 meter	47

DAFTAR GAMBAR

Gambar 2.1 Gelombang Suara	6
Gambar 2.2 Arduino Uno Pinout.....	7
Gambar 2.3 Komunikasi Arduino Uno	9
Gambar 2.5 ESP8266 WIFI Module	10
Gambar 2.6 Bluetooth module	10
Gambar 2.7 NRF pinout.....	11
Gambar 3.1 Alur Metodologi Penelitian	12
Gambar 3.2 alur kerja pada pengirim	14
Gambar 3.3 alur kerja pada penerima	14
Gambar 5.1 Diagram Blok Perancangan Sistem	20
Gambar 5.2 Perancangan Transmitter pada NRF.....	21
Gambar 5.3 Perancangan Transmitter pada Bluetooth	22
Gambar 5.4 Perancangan Transmitter pada Wifi.....	23
Gambar 5.5 Perancangan Reciever pada NRF	24
Gambar 5.6 Perancangan Reciever pada Bluetooth.....	25
Gambar 5.7 Perancangan Reciever pada Wifi	26
Gambar 5.8 Perancangan Kalibrasi modul nrf	27
Gambar 5.9 Perancangan Kalibrasi modul Bluetooth.....	27
Gambar 5.10 Perancangan Kalibrasi modul wifi	28
Gambar 5.11 Implementasi transmitter nrf	29
Gambar 5.12 Implementasi transmitter Bluetooth	30
Gambar 5.13 Implementasi transmitter Wifi	30
Gambar 5.14 Implementasi reciever nrf	31
Gambar 5.15 Implementasi reciever Bluetooth	31
Gambar 5.16 Implementasi reciever Wifi	32
Gambar 6.1 Percobaan nrf dengan selisih 1 meter	37
Gambar 6.2 Percobaan nrf dengan selisih 3 meter	38
Gambar 6.3 Percobaan nrf dengan selisih 6 meter	39
Gambar 6.4 Percobaan modul HC-05 dengan selisih 1 meter.....	40
Gambar 6.5 Percobaan modul HC-05 dengan selisih 3 meter.....	41

Gambar 6.6 Percobaan modul HC-05 dengan selisih 6meter.....	42
Gambar 6.7 Percobaan modul ESP8266 dengan selisih 1 meter.....	43
Gambar 6.8 Percobaan modul ESP8266 dengan selisih 3 meter.....	44
Gambar 6.9 Percobaan modul ESP8266 dengan selisih 6 meter.....	45

DAFTAR LAMPIRAN

LAMPIRAN A Badan Program	51
A. Program Bluetooth	51
B. Program Wifi	52

BAB 1 PENDAHULUAN

1.1 Latar belakang

Semakin berkembang secara luas teknologi mengakibatkan pentingnya peranan media elektronik dalam penyampaian informasi dalam kehidupan sehari-hari. Terdapat berbagai media untuk menyampaikan informasi baik secara digital, gambar, dan audio.

Teknologi sekarang banyak mengalami perkembangan. Dan perkembangan teknologi sangat mempengaruhi berbagai aspek di Kehidupan kita, dan untuk fokus pada perkembangan teknologi audio (Grant, A. E. & Meadows, J. H. 2010). Audio merupakan penyampai informasi yang berupa sinyal suara. dengan berkembangnya teknologi sekarang, tidak jarang audio digunakan sebagai media penyampaian pesan.

Dalam mendapatkan kualitas suara yang baik perlu diperhatikan beberapa aspek yang mempengaruhi kualitas suara seperti media penghubung dan penempatan sumber suara dan penerima suara. pada beberapa kasus untuk pengiriman suara masih menggunakan kabel, jadi ada kalanya terdapat keterbatasan media penyampaian suara.

Terdapat banyak media sebagai penghantar suara secara wireless, seperti wifi, Bluetooth, dan nrf. Penulis ingin membandingkan pengiriman suara melalui ketiga media tersebut. Diharapkan media yang paling cepat mengantarkan sinyal suara bisa digunakan secara optimal dalam kehidupan sehari-hari.

Sebuah penelitian yang berjudul *Voice Transmission over Wireless Sensor Networks* membahas tentang pengiriman suara melalui wireless sensor network. Dalam menggunakan jaringan WSN, peneliti mengirimkan suara memakai zigbee dan Bluetooth (Chang, Chen, & Lee 2012).

Penelitian lain dengan judul *Wireless Voice Transmission using WIFI and Bluetooth on Android Platform* membahas tentang perancangan android untuk transmisi suara secara wireless. Media yang digunakan ada dua yaitu wifi dan bluetooth. (Shinkar, Shinde, Goswami, & Shinde 2018).

Pada penelitian sebelumnya hanya membandingkan dua jenis media pengiriman suara. Dalam penelitian sebelumnya juga, ada penelitian yang dilakukan dengan memakai suara sintesis, artinya data yang digunakan merupakan data statis yang sebelumnya sudah ditetapkan.

Berdasarkan beberapa penelitian diatas, maka penulis akan mencoba untuk mendesain perangkat pengirim dan penerima suara secara realtime menggunakan ketiga jenis jaringan wireless yaitu wifi, Bluetooth, dan radio. Alat ini akan menggunakan Modul nRF24L01 sebagai media pengiriman suara melalui radio. Modul HC-02 sebagai media pengiriman suara melalui bluetooth. Dan modul ESP8266 sebagai media pengiriman suara melalui wifi. Untuk mikrokontroler menggunakan Arduino Uno. Masing – masing pengiriman suara

akan dirancang kedalam alat transmitter dan receiver. Diharapkan dari hasil penelitian ini dapat menentukan media pengiriman suara yang terbaik.

1.2 Rumusan masalah

Pada Implementasi Perbandingan Sistem Pengiriman Suara Secara Wireless Menggunakan Arduino, penulis membatasi permasalahan dari perancangan sistem, meliputi :

1. Bagaimana cara merancang sistem pengiriman sinyal suara secara wireless?
2. Bagaimanakah pengaruh faktor yang terlibat dalam pengiriman suara secara wireless?
3. Bagaimanakah hasil yang didapat dari performa masing masing media yang digunakan sebagai pengiriman sinyal suara ?

1.3 Tujuan

Ada pun Tujuan pembuatan skripsi Implementasi Perbandingan Sistem Pengiriman Suara Secara Wireless meliputi:

1. Membuat sistem pengiriman suara dan penerimaan secara wireless.
2. Membuat alur kerja pengiriman data berupa suara.
3. Membandingkan performa masing masing media wireless yang digunakan.

1.4 Manfaat

Manfaat yang bisa didapatkan dari penelitian Implementasi Sistem Pengiriman Suara Secara Wireless adalah :

1. Peneliti dapat mengetahui batas-batas pengiriman sinyal suara
2. Dapat menyesuaikan jenis wireless yang digunakan
3. Mengetahui jenis jaringan yang terbaik untuk pengiriman data secara wireless
4. Manfaat untuk kedepannya bisa digunakan untuk penelitian project selanjutnya.

1.5 Batasan masalah

Agar sistem yang dibuat terarah dan tidak melenceng maka dibuat batasan masalah yang meliputi :

1. Perancangan sistem pengiriman suara yang menggunakan arduino uno sebagai pemroses data.
2. Menggunakan sinyal suara sebagai pengiriman data
3. Jaringan wireless yang digunakan adalah jaringan wifi, Bluetooth, dan nrf.
4. Menggunakan 2 buah alat yang dapat digunakan sebagai satu node pengirim dan node penerima
5. Dalam pembuatan sistem ini lebih difokuskan pengiriman sinyal suara ke masing masing node dengan berbasis wireless

6. Akan dilakukan analisa terhadap masing masing jaringan yang digunakan, untuk menentukan jaringan manakah yang lebih bagus dalam pengiriman sinyal suara

1.6 Sistematika pembahasan

Dalam penyusunan laporan, penulis membuat laporan berdasarkan pada sistematika pembahasan yang dijabarkan pada penjelasan berikut:

BAB I PENDAHULUAN

Dalam menyusun laporan, penulis membuat latar belakang berdasarkan masalah yang ada, kemudian membuat rumusan masalah sebagai acuan dari masalah yang dijabarkan. Penulis juga membuat tujuan dan manfaat dari dilakukannya penelitian serta batasan masalah agar penelitian bisa dilakukan dengan terarah.

BAB 2 LANDASAN KEPUSTAKAAN

Penulis mencoba membandingkan penelitian sebelumnya dengan penelitian yang sedang dikerjakan. Dalam bab ini juga memuat landasan pustaka yang berisi teori penulis dalam melakukan pengembangan penelitian.

BAB 3 METODOLOGI PENELITIAN

Dalam melakukan penelitian, penulis mencoba membuat rencana alur penelitian. Untuk itu dilakukannya metodologi pada penelitian. Pertama membuat diagram alur yang berisi tahapan penulis dalam melakukan penelitian. Kemudian menjelaskan studi literatur yang digunakan secara umum. Untuk perencanaan apa yang dibutuhkan dalam penelitian, maka dituangkan pada rekayasa kebutuhan sistem. Setelah itu membuat rancangan alat agar bisa bekerja, bisa berupa rancangan pada perangkat keras maupun pada perangkat lunak. Setelah apa yang semua dibutuhkan siap, maka dilakukan implementasi berdasarkan rancangan yang dibuat. Tahap selanjutnya dilakukannya pengujian dan analisa dari alat yang telah dibuat. Dan terakhir membuat kesimpulan dari analisa sistem.

BAB 4 REKAYASA KEBUTUHAN

Rekayasa kebutuhan digunakan sebagai acuan penulis untuk menjelaskan kebutuhan yang diperlukan dalam melakukan penelitian. Pada rekayasa kebutuhan, kebutuhan yang diperlukan secara umum dibagi menjadi dua yaitu kebutuhan sistem dan kebutuhan antarmuka. Dalam kebutuhan sistem, terdapat beberapa bagian kebutuhan yang diperlukan. Pertama kebutuhan perangkat keras meliputi mikrokontroler dan modul yang akan digunakan. Kebutuhan perangkat lunak meliputi program dan perangkat lunak yang digunakan. Serta kebutuhan fungsional dan non fungsional yang dibutuhkan penelitian. Dalam kebutuhan antarmuka, terdapat kebutuhan antarmuka dengan pengguna, dengan perangkat keras, dan dengan perangkat lunak.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada perancangan dan implementasi, penulis menjelaskan tentang rancangan pada alat, dapat berupa rancangan untuk perangkat keras yang dijelaskan dalam skema rangkaian dan rancangan perangkat lunak yang dijelaskan pada flowchart. Untuk implementasi berupa pemasangan perangkat keras sesuai dengan skema rancangan dan eksekusi program pada sistem.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini penulis melakukan uji coba pada alat yang sudah jadi, uji coba dilakukan dengan beberapa tahap yaitu uji coba terhadap masing-masing alat dan ujicoba dengan menggunakan semua alat. Dalam melakukan uji coba, penulis juga melakukan pencatatan data yang didapatkan. Kemudian dari data tersebut dibuat analisa yang dijelaskan kedalam tabel dan penjelasan.

BAB 7 PENUTUP

Bab penutup berisi dua sub bab, yaitu kesimpulan. Kesimpulan berisi jawaban atas rumusan masalah yang ada. Jawaban tersebut didapatkan dengan melakukan pengujian dan analisis. Kemudian sub bab saran yang berisi saran pengembangan penelitian ini untuk kedepannya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Sebuah penelitian yang berjudul *Voice Transmission over Wireless Sensor Networks* membahas tentang pengiriman suara melalui wireless sensor network. Dalam menggunakan jaringan WSN, peneliti mengirimkan suara memakai zigbee dan Bluetooth (Chang, Chen, & Lee 2012).

Penelitian lain dengan judul *Wireless Voice Transmission using WIFI and Bluetooth on Android Platform* membahas tentang perancangan android untuk transmisi suara secara wireless. Media yang digunakan ada dua yaitu wifi dan bluetooth. (Shinkar ,Shinde , Goswami , & Shinde 2018).

Penelitian yang berjudul Perancangan sistem komunikasi Dua Arah Dengan Arduino Pada Frekuensi 2,4 Ghz membahas tentang perancangan alat yang bisa mengirimkan suara dan menerima suara secara wireless. Alat ini menggunakan modul NRF24L01 sebagai modul pengiriman.

Pada penelitian ini akan merancang pengiriman suara melalui wireless network dengan memakai tiga module, yaitu modul Nrf24l01 sebagai pengiriman suara melalui radio, module HC-05 sebagai pengiriman suara melalui bluetooth, dan module ESP8266 sebagai pengiriman suara melalui wifi.

Tabel 2.1 Perbandingan Penelitian

NO	Nama penulis, tahun, judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	(Chang, Chen, & Lee 2012). <i>Voice Transmission over Wireless Sensor Networks</i>	Membahas tentang pengiriman suara melalui wireless sensor network	Membahas tentang pengiriman suara secara wireless dengan menggunakan suara sintesis.	Membahas tentang perancangan pengiriman suara realtime secara wireless
2	(Shinkar ,Shinde , Goswami , & Shinde 2018. <i>Wireless Voice Transmission using WIFI and Bluetooth on Android</i>	Menggunakan media yaitu wifi dan bluetooth sebagai perbandingan	Membahas tentang pengiriman suara secara wireless dengan menggunakan android	Menggunakan media Nrf, Bluetooth, dan wifi Perbandingan.

	<i>Platform</i>			
3	(Guntur, Purwandi, & Rasyid 2018) Perancangan sistem komunikasi Dua Arah Dengan Arduino Pada Frekuensi 2,4 Ghz	Menggunakan Nrf24L01 sebagai modul pengiriman.	Penelitian terdahulu hanya merancang satu modul pengiriman	Merancang tiga modul pengiriman agar bisa dilakukan perbandingan.

2.2 Dasar Teori

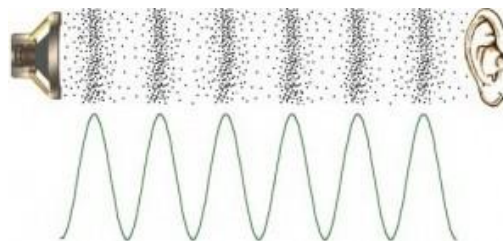
Dasar teori merupakan informasi terkait penelitian ini yang dikumpulkan dari berbagai sumber. Kajian informasi ini digunakan sebagai pendukung dalam merancang alat.

2.2.1 Sinyal Audio

Audio pada dasarnya adalah gelombang yang bisa didengar oleh telinga kita. Gelombang suara itu bisa disebut juga sinyal audio. Gelombang suara bisa berbeda-beda karena beberapa faktor. Salah satunya adalah udara, karena suara adalah gelombang yang bergetar dan melalui perantara udara, maka udara juga ikut bergetar yang mengakibatkan gendang telinga kita ikut bekerja.

Pada umumnya manusia hanya bisa mendengar suara dengan frekuensi 20 Hz sampai dengan 20000 Hz. 20 Hz merupakan frekuensi terendah yang bisa kita dengar, dan 20000Hz merupakan frekuensi paling tinggi yang bisa kita dengar. (Teori Elektronika, 2018).

Dapat dilihat pada gambar 2.1 merupakan ilustrasi suara sampai ke telinga kita. Awalnya speaker bergetar dan menghasilkan gelombang. Gelombang ini kemudian merambat melalui udara disekitar kita. Gelombang ini dapat juga digambarkan dengan garis panjang yang melengkung .



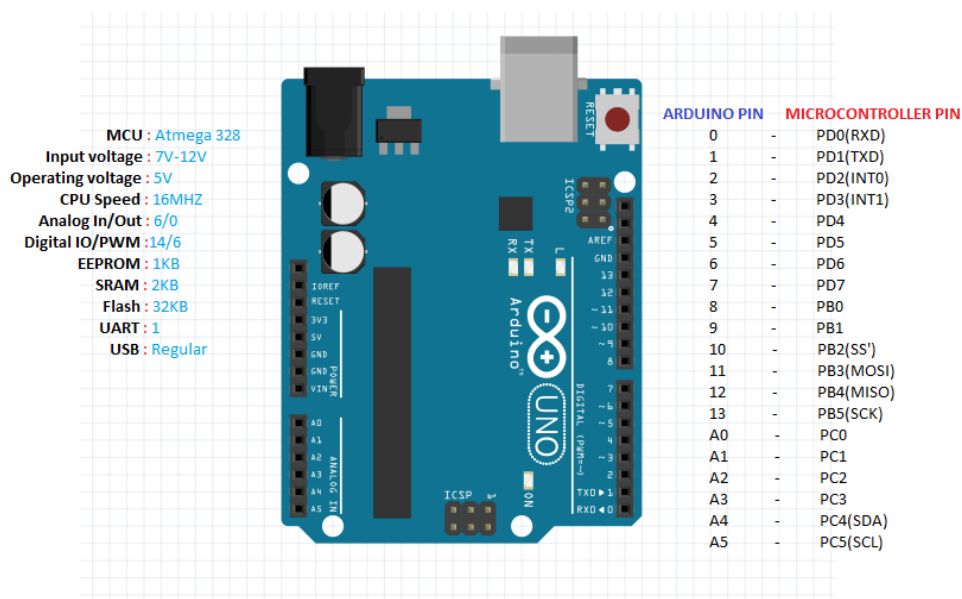
Gambar 2.1 Gelombang Suara
Sumber: (Teori Elektronika, 2018)

Titik-titik hitam yang ada pada gambar 2.1 jumlahnya banyak dapat kita sebut molekul pada udara. Saat speaker menghasilkan suara maka molekul suara tersebut akan bergetar dan bisa ditunjukkan kedalam gelombang,. Saat gelombang suarai ini sampai ke telinga kita, maka gendang telinga akan bekerja dengan mengikuti getaran gelombang mengikuti pola.

Setiap molekul udara berpindah pada jarak yang kecil sebagai getaran, namun mengakibatkan molekul yang bersebelahan bergetar semua terpengaruh berjalan sampai telinga. Itulah proses Molekul udara sampai dari loudspeaker ke telinga kita. Semua gelombang pasti memiliki tiga sifat penting untuk kerja audio meliputi : panjang gelombang, amplitudo dan frekuensi.

2.2.2 ARDUINO UNO

Arduino Uno merupakan mikrokontroller yang menggunakan Atmega328p sebagai chip utama. Pada mikrikontroller ini, umumnya mempunyai fungsi sebagai papan prototype dalam rangkaian project. Penggunaan Arduino Uno sebagai mikrokontroler sangat direkomendasikan karena chip ATmega328p sudah terhubung dengan papan sirkuit dan pinout sehingga memeudahkan kita dalam melakukan pengembangan alat dibandingkan dengan merangkai ATmega328p secara manual pada papan board dan melakukan solder secara manual. (Arduino,2016)



Gambar 2.2 Arduino Uno Pinout

Sumber: (Arduino,2016)

Gambar 2.3 menunjukkan papan Arduino Uno dan pinout pada papan. Untuk pin input Arduino Uno mempunyai 14 pin digital yang bisa digunakan. Sedangkan untuk pin output, Arduino Uno mempunyai 6 pin output. Untuk pin analog terdapat 6 pin. Selain pinout pada Arduino Uno juga terdapat port untuk koneksi pada kabel jack yang bisa digunakan untuk adaptor, sebuah tombol reset yang berfungsi untuk mereset program pada board, dan Port USB untuk komunikasi

program pada komputer. Penggunaan Arduino Uno mempunyai keuntungan karena jika terjadi kerusakan maka kita hanya perlu mengganti chip ATmega328, tidak perlu mengganti seluruh board.

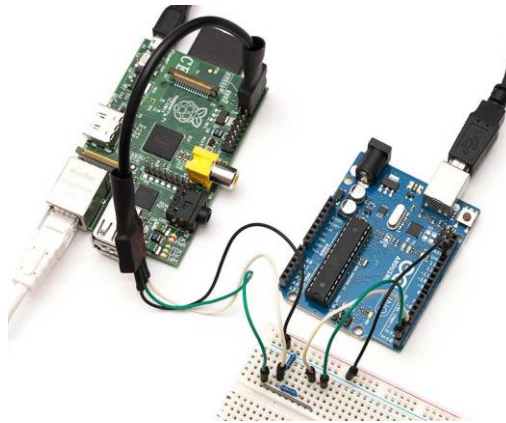
Terdapat 14 pin digital yang ada pada Arduino Uno. Pada setiap pin dapat mempunyai perintah untuk menjalankan fungsi pinMode, fungsi untuk menggunakan pin sebagai pin input maupun pin output, fungsi yang kedua adalah menjalankan perintah digitalWrite, yaitu fungsi pada pin untuk menulis data digital, dan yang terakhir fungsi digitalRead, yaitu fungsi pin untuk membaca data digital yang biasa diterima oleh sensor tambahan. Setiap pin membutuhkan tegangan 5V dan arus 20mA, dan juga mempunyai batas 40mA. Jika tegangan melewati batas tersebut dikhawatirkan akan terjadi kerusakan pada chip ATmega328.

Beberapa pin memiliki fungsi khusus :

- **External Interrupts**, terdapat pada pin 2 dan pin 3. Pin tersebut biasanya digunakan untuk mengaktifkan perintah interrupts.
- **Serial**, pada serial terdapat 2 pin : pin 0 (RX) dan pin 1 (TX) digunakan dalam menerima (RX) dan pengiriman (TX) data serial.
- **SPI** : Untuk komunikasi SPI pada SPI Library menggunakan Pin 13 (SCK), pin 10 (SS), pin 12 (MISO), dan pin 11 (MOSI).
- **LED** : untuk LED menggunakan Pin 13. Pin 13 juga sudah terhubung built-in led yang menggunakan perintah digital pin no 13.
- **I2C** : Dalam komunikasi I2C pada Wire Library menggunakan Pin A4 (SDA) dan pin A5 (SCL).
- **PWM**: Pin 3, 5, 6, 9, 10, dan 11 menyediakan output PWM 8-bit dengan menggunakan fungsi analogWrite()

Pada mikro kontroler Arduino Uno terdapat 6 buah input analog, yaitu pin A0, pin A1, pin A2, pin A3, pin A4, dan pin A5. Pada setiap pin analog tersebut terdapat resolusi 10 bits. Pada dasarnya, pin tersebut terhubung melalui ground ke daya 5V, untuk alternatif lain pin AREF juga dapat digunakan sebagai fungsi analogReference(). Beberapa input lainnya pada board ini adalah :

- **AREF**. Digunakan untuk mengatur tegangan terhadap input analog.
- **Reset**. Digunakan untuk melakukan reset terhadap mikrokontroler. Fungsi ini Sama dengan pemakaian tombol reset.



Gambar 2.3 Komunikasi Arduino Uno

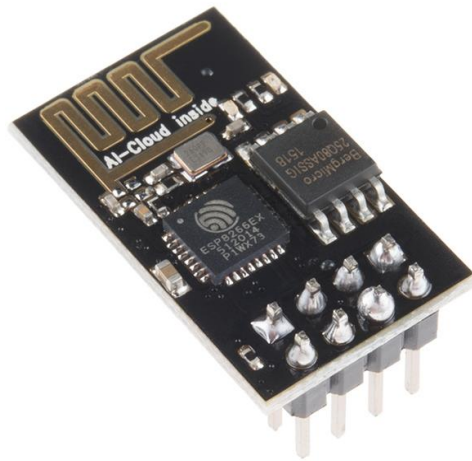
Sumber: (Escadio, 2017)

Pada gambar 2.4 dapat dilihat contoh bagaimana Arduino Uno berkomunikasi. Untuk melakukan komunikasi pada serial bus, kita memerlukan port USB. Kita juga dapat menggunakannya untuk berkomunikasi dengan sesama arduino maupun dengan mikro kontroler lain. Pada papan Arduino, Chip ATmega16U2 mempunyai fungsi untuk penterjemah komunikasi USB saat kita menghubungkan Arduino dengan komputer. Dengan fungsi ini sudah termasuk driver USB standar, maka saat kita melakukan komunikasi Arduino dengan komputer kita tidak perlu melakukan instalasi driver tambahan.

Dalam melakukan komunikasi Arduino dengan komputer, kita memerlukan perangkat lunak yang bernama Arduino Software (IDE). Perangkat lunak ini berfungsi untuk melakukan eksekusi program yang ditujukan pada Arduino. Jika program telah terkirim pada Arduino maka led TX maupun led RX akan menyala secara konstan. Pengiriman program dilakukan dengan menggunakan port USB pada papan Arduino yang dihubungkan dengan komputer. Selain mendukung komunikasi serial bus, Arduino juga mendukung komunikasi I2C dan SPI (Escadio, 2017).

2.2.3 ESP8266 WiFi Module

Modul WIFI ESP8266 merupakan modul yang berupa sistem dalam bentuk chip. Dengan menggunakan modul ini pada suatu mikro kontroler, maka mikro kontroler tersebut dapat terhubung dengan jaringan WIFI, karena modul ini dapat terintegrasi dengan protokol internet IP/TCP. Pada jaringan WIFI modul ESP8266 dapat berfungsi sebagai host maupun client. Untuk penggunaan modul ESP8266 dengan Arduino, kita bisa melakukan pengiriman data secara wireless pada jaringan WIFI maupun sebagai WIFI shield pada Arduino. Karena Modul ESP8266 merupakan salah satu board yang memiliki komunitas pengguna yang besar, maka setiap tahun modul ini akan selalu berkembang (ESP8266, 2014). Pada gambar 2.5 berikut kita bisa melihat ilustrasi modul ESP8266.



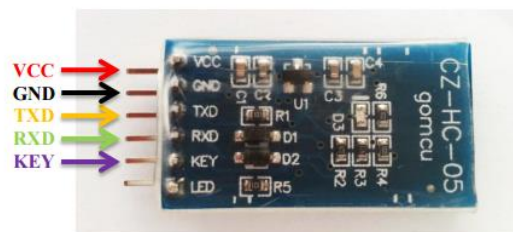
Gambar 2.4 ESP8266 WIFI Module

Sumber: (ESP8266, 2014)

Dibanding modul WIFI sejenisnya, modul ESP8266 memiliki keunggulan dalam melakukan pengolahan dan penyimpanan data. Dengan keunggulan tersebut kita bisa menggunakan modul ini agar terhubung dengan sensor maupun perangkat lain dengan menggunakan GPIOs. Karena modul ini dibuat seminimal mungkin, maka modul ini dapat kita rancang semaksimal mungkin dalam keterbatasan letak PCB. Modul ESP8266 mendukung RSDP untuk aplikasi VoIP dan Bluetooth interface co-eksistensi, berisi RF dikalibrasi memungkinkan untuk bekerja di bawah semua kondisi operasi, dan tidak memerlukan bagian RF eksternal.

2.2.4 HC-05 Bluetooth Modul

HC-05 Bluetooth Module merupakan modul yang menggunakan Bluetooth SPP modul (Serial Port Protocol). Modul ini dirancang untuk pengaturan koneksi serial nirkabel secara transparan. komunikasi dilakukan melalui komunikasi serial yang terhubung secara antarmuka dengan controller atau PC (HC-05, 2010). Modul HC-05 Bluetooth menyediakan modus beralih antara mode pengirim dan mode penerima yang berarti dapat menerima atau mengirimkan data. Dalam gambar 2.6 kita bisa melihat ilustrasi module HC-05 dan pin-pin yang digunakan.



Gambar 2.5 Bluetooth module

Sumber: (HC-05, 2010).

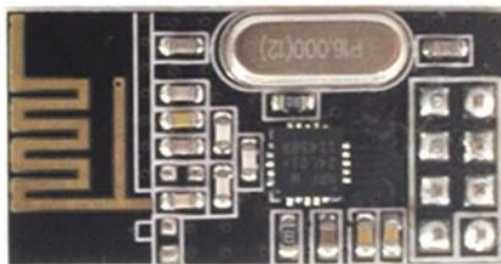
Pada gambar 2.6 kita bisa melihat pin yang ada pada modul HC-05. Modul HC-05 menggunakan 5 pin. Pin Vcc digunakan sebagai pin untuk sumber daya

modul ini, yang biasanya membutuhkan daya 3,3V. TDX merupakan pin yang digunakan untuk pengiriman data melalui bluetooth, sedangkan RDX merupakan pin yang digunakan untuk penerimaan data melalui bluetooth. Kedua pin tersebut biasanya digunakan pada Arduino dan mikrokontroler lainnya. Pin KEY dapat digunakan untuk mode switch.

2.2.5 NRF24L01

Modul NRF24L01 merupakan modul berbasis sinyal radio. Modul ini merupakan modul yang umumnya digunakan dalam pengiriman data secara wireless. Modul NRF24L01 menggunakan gelombang radio pada frekuensi 2,4 GHz.

Modul RF24L01 merupakan modul komunikasi yang banyak diminati orang-orang. karena menggunakan daya kecil maka modul ini bisa digunakan bertahun-tahun. Modul ini biasa digunakan untuk pembuatan ptheriperal PC, piranti permainan, piranti fitnes dan olahraga, mainan anak-anak dan alat lainnya (Nrf24l01, 2014).



Gambar 2.6 NRF pinout

Sumber: (Nrf24l01, 2014)

Pada gambar 2.7 dapat dilihat pin yang ada pada modul NRF24L01. Modul NRF24L01 terdapat 8 buah pin, yaitu sebagai berikut :

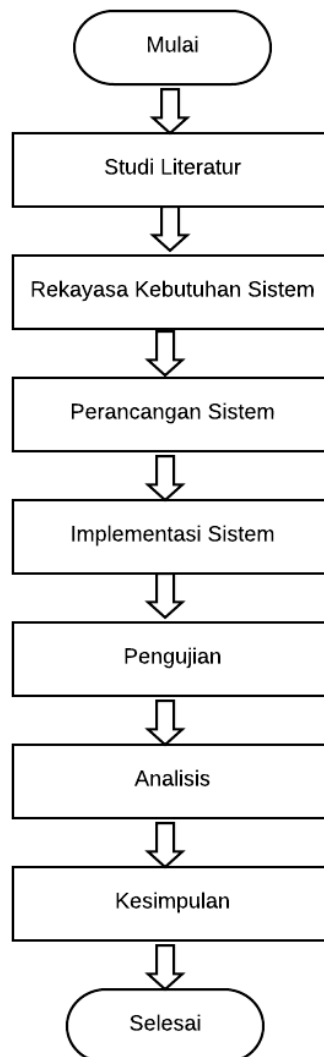
1. VCC (3.3V DC)
2. Pin GND
3. Pin CSN
4. Pin IRQ
5. Pin MISO
6. Pin SCK
7. Pin MOSI
8. Pin CE

Modul NRF24L01 mempunyai beberapa keunggulan yaitu penggunaan daya yang bisa terbilang kecil, untuk pengiriman data bisa maupun penerimaan data bisa dilakukan secara otomatis. Serta pengiriman data yang bisa mencapai rate 2 Mbps.

BAB 3 METODOLOGI

3.1 Alur Metodologi Penelitian

Pada alur metode penelitian, penulis menjabarkan tentang bagaimana penelitian ini dikerjakan. Alur pengerjaan penelitian dibuat dalam diagram alur agar mudah dimengerti. Pada gambar 3.1 merupakan gambaran metodologi secara umum yang dikerjakan penulis.



Gambar 3.1 Alur Metodologi Penelitian

Penulis mulai melakukan penelitian dengan mencari masalah dalam kehidupan sehari-hari. Setelah menemukan masalah yang bisa dijadikan penelitian, maka penulis melakukan studi literatur yang membahas masalah tersebut dan teori untuk mendukung menyelesaikan masalah tersebut. Kemudian penulis melakukan rekayasa kebutuhan, mencatat apa yang dibutuhkan dalam melakukan penelitian. Langkah selanjutnya adalah membuat rancangan alat yang bisa digunakan, dapat berupa rancangan perangkat keras

maupun perangkat lunak. Setelah semua yang dibutuhkan sekiranya lengkap, penulis melakukan implementasi alat baik untuk perangkat keras maupun perangkat lunak. Setelah alat telah jadi, maka penulis melakukan pengujian dan analisa dari percobaan alat tersebut. Dan tahap terakhir penulis akan membuat suatu kesimpulan dan penyelesaian masalah dari rumusan masalah yang telah dibuat.

3.2 Studi Literatur

Pada tahap ini penulis merangkum teori yang dibutuhkan dalam melakukan perancangan alat pengiriman suara secara wireless ini. Beberapa teori yang didapatkan dari berbagai sumber dimasukkan dalam studi literatur, berikut dasar teori acuan yang digunakan untuk implementasi alat pengiriman suara secara wireless :

1. Teori tentang suara

Mempelajari tentang suara dan bagaimana pengaplikasiannya pada penelitian.

2. Mikrokontroler Arduino Uno

Mempelajari apa itu Arduino Uno, bagaimana fungsinya dan implementasinya sebagai mikrokontroler perangkat utama dari sistem pengiriman suara. (Sumber: Arduino, 2016)

3. Modul nRF24L01

Mempelajari cara kerja modul *NRF24L01* dalam mengirimkan data serta pin yang digunakan. (Sumber: Nrf24l01, 2014)

4. Modul bluetooth

Mempelajari bagaimana modul *Bluetooth* bekerja, pinout yang digunakan serta bagaimana mengirimkan data. (Sumber: HC-05 2010)

5. Modul wifi

Mempelajari bagaimana modul *wifi* bekerja, pin yang digunakan serta bagaimana mengirimkan data. (Sumber: ESP8266, 2014)

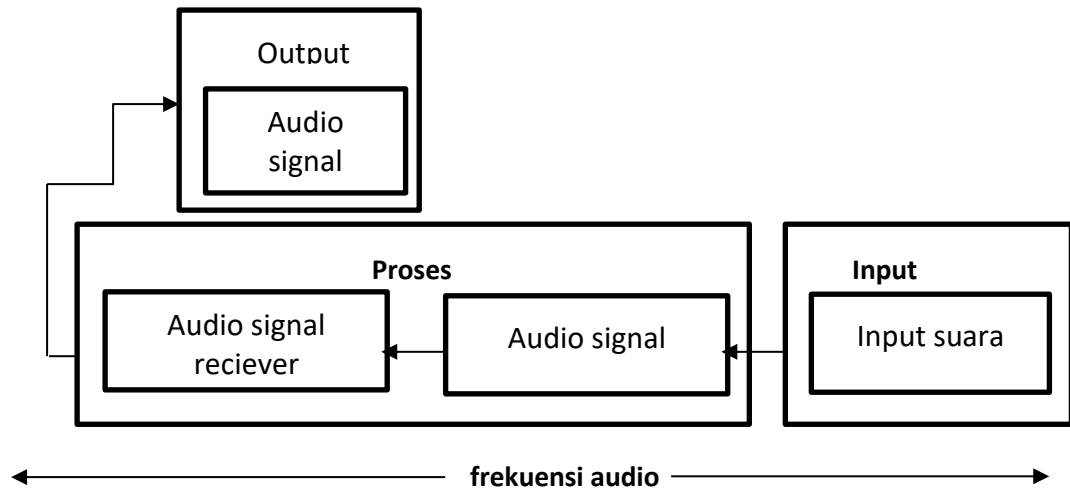
3.3 Rekayasa Kebutuhan Sistem

Dibuatnya rekayasa kebutuhan sistem bertujuan untuk perancangan apa yang dibutuhkan oleh alat pengiriman suara yang dibuat agar penulis dapat mempermudah dalam merancang sistem tersebut.

Penulis melakukan analisa yang dibutuhkan sistem sebagai acuan untuk mengetahui hal apa yang diperlukan dalam implementasi sistem pengiriman suara secara wireless. Dalam melakukan analisis kebutuhan penulis mencoba mengidentifikasi masalah yang ada, kemudian menentukan tahap penelitian selanjutnya, baik untuk pembuatan perangkat keras maupun pembuatan perangkat lunak.

3.4 Perancangan Sistem

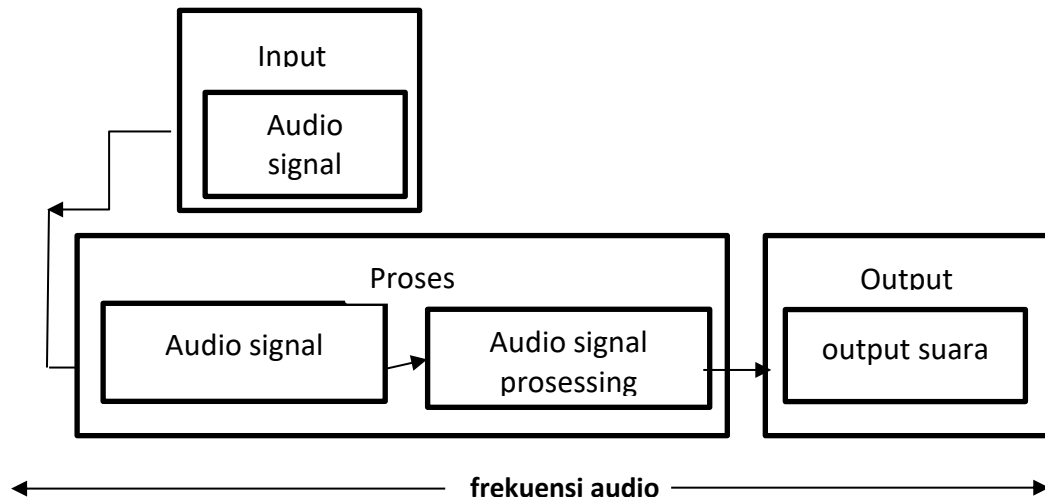
Pada tahap ini, penulis menjabarkan perancangan sistem yang dibagi kedalam tiga tahap yaitu Input, proses, dan output. Perancangan alat pengiriman suara secara wireless ini meliputi masukan berupa sinyal suara, mikrokontroler Arduino Uno untuk mengolah pengiriman data yang berupa sinyal suara.



Gambar 3.2 Alur kerja Pada Pengirim

Pada gambar ditunjukkan alur kerja pengirim. Pengguna menginput suara yang berupa audio signal, kemudian audio signal akan diproses untuk dikirimkan secara wireless.

Untuk bagian penerima digambarkan dalam alur kerja pada gambar 3.3 berikut.



Gambar 3.3 Alur kerja Pada Penerima

Setelah dilakukan pengiriman, maka data yang diterima berupa audio signal. Pada tahap proses Audio signal diproses oleh mikrokontroler untuk selanjutnya di bunyikan melalui speaker secara real time.

3.4.1 Perancangan perangkat keras

Penulis membagi tiga jenis modul perancangan perangkat keras yaitu untuk pengiriman melalui wifi, Bluetooth, dan nrf. Pada masing- masing modul pengiriman di rancang sebuah transmitter sebagai pengiriman data. Untuk bagian penerima menggunakan arduino sebagai pemroses sinyal suara yang diterima oleh module.

3.4.2 Perancangan perangkat lunak

Agar alat bisa beroperasi dengan yang diharapkan maka dilakukan perancangan perangkat lunak. Penulis membuat flowchart berdasarkan jalan program yang diharapkan agar bisa membuat coding program yang bisa digunakan pada transmitter dan receiver untuk menjalankan sistem pada perangkat keras.

3.5 Implementasi Sistem

Setelah dilakukan perancangan, maka tahap selanjutnya melakukan implementasi yang dibagi dalam dua tahap yaitu:

1. Implementasi perangkat keras, penulis melakukan pembuatan alat berdasarkan rancangan yang dibuat. Dalam implementasi perangkat keras, dilakukan pemasangan pin pada mikrokontroler dan modul serta perangkat keras lain.
2. Implementasi perangkat lunak, penulis menggunakan program Arduino IDE untuk mengunggah program pada mikrokontroler. Program tersebut dibuat berdasarkan flowchart yang telah dibuat sebelumnya.

3.6 Pengujian dan Analisis

Dalam Pengujian sistem, terdapat beberapa uji coba alat yang berupa pengujian dengan Nrf, pengujian dengan bluetooth, pengujian dengan wifi dan pengujian modul secara keseluruhan.

3.6.1 Pengujian nRF24L01

Pengujian dilakukan menggunakan modul nRF24L01 pada tempat tenang. Nilai hasil pengujian ini dicatat untuk kemudian dibandingkan.

3.6.2 Pengujian Bluetooth

Pengujian modul bluetooth dan membandingkan hasilnya.

3.6.3 Pengujian Wifi

Pengujian modul Wifi. Pengujian ini dilakukan untuk mendapat nilai data agar bisa dilakukan analisa pengujian.

3.6.4 Pengujian Keseluruhan sistem

Pada pengujian ini penulis merangkum ketiga pengujian sebelumnya. Dari ketiga pengujian tersebut maka akan di buat perbandingan nilai yang didapat

dari masing masing modul. Selanjutnya data yang didapatkan akan dicatat dan dibandingkan untuk mendapatkan kesimpulan analisa.

3.7 Kesimpulan

Setelah melihat analisa yang telah dibuat, penulis mencoba membuat kesimpulan. Kesimpulan tersebut dibuat berdasarkan percobaan yang telah dilakukann dengan rumusan masalah yang ada.

BAB 4 REKAYASA KEBUTUHAN

4.1 Deskripsi Umum

Perancangan sistem pengiriman suara ini merupakan perangkat yang dapat mengirimkan suara serta menerima suara yang diterima. Alat ini menggunakan arduino uno sebagai mikro kontroler utama, modul nRF24L01, modul bluetooth, serta modul wifi.

Dalam perancangan sistem pengiriman suara secara wireless, Arduino Uno digunakan sebagai mikrokontroler dengan berbagai pertimbangan, seperti harga yang terjangkau. Ketiga modul menggunakan mikrokontroler yang sama.

Masukan sistem ini menggunakan tiga modul. Pembacaan dari modul dapat berupa nilai data suara yang diterima oleh modul receiver.

4.1.1 Perspektif sistem

Sistem dapat melakukan pengiriman suara. Selain itu ada juga penerimaan suara. Sistem ini mengirimkan data suara secara realtime dan ditampilkan dalam output berupa suara.

4.1.2 Kegunaan Sistem

Pada tahap ini Sistem dapat digunakan sebagai alat uji coba untuk penelitian pengiriman suara. Untuk transmitter akan terhubung sesuai modul yang akan diuji coba. Selain itu, sistem ini juga terdapat receiver yang berfungsi sebagai penerimaan data.

4.1.3 Karakteristik pengguna

Penggunaan alat pengiriman suara secara wireless hanyalah sebagai alat pengujian, untuk itu dalam melakukan percobaan pada sistem maka dibutuhkan suara pengguna sebagai masukan data pada sistem.

4.1.4 Lingkungan Operasi

Dalam melakukan pengujian alat pengiriman suara secara wireless, penulis melakukan operasi alat pada tempat sunyi dan tenang. Karena suara pada lingkungan dapat menimbulkan noise berlebihan.

4.1.5 Batasan Sistem

Penulis melakukan penelitian berdasarkan batasan untuk perancangan dan implementasi yang mencakup

1. Mikro kontroler utama menggunakan Arduino Uno.
2. Menggunakan modul nRF24L01, bluetooth, dan wifi sebagai modul pengirim dan penerima data.
3. Menggunakan aplikasi sndpeek sebagai penampil data suara.

4.2 Rekayasa Kebutuhan Sistem

Dalam membuat alat pengiriman suara secara wireless, penulis memerlukan berbagai media yang diperlukan. Terdapat beberapa kebutuhan yang dijabarkan pada penjelasan berikut.

4.2.1 Kebutuhan perangkat keras

Pada pembuatan alat pengiriman suara secara wireless, dibutuhkan berbagai perangkat keras yang nantinya bisa digunakan sesuai batasan masalah. Perangkat keras ini meliputi Arduino Uno, NRF24L01, HC-02, dan ESP8266.

4.2.1.1 Arduino Uno

Dalam merancang alat pengiriman suara secara wireless, Arduino Uno dipilih karena berdasarkan jumlah *port input* dan *output* yang dibutuhkan untuk modul yang digunakan. Dalam sistem ini sebuah arduino uno dibutuhkan sebagai masukan badan program. Selain itu untuk perbandingan data maka digunakan mikro kontroler yang sama pada ketiga modul pengiriman data.

4.2.1.2 nRF24L01

Modul nRF24L01 digunakan sebagai pengiriman suara dengan media radio. Karena harga yang murah dan mudah didapatkan.

4.2.1.3 Bluetooth

Penggunaan modul HC-02 sebagai pengiriman melalui media bluetooth digunakan karena penggunaan pin yang sesuai dengan arduino uno. Selain itu modul ini mudah didapat dengan harga yang terjangkau.

4.2.1.4 wifi

Penggunaan modul ESP8266 sebagai pengiriman melalui media wifi karena penggunaan pin karena modul ini mudah didapat dengan harga yang terjangkau.

4.2.2 Kebutuhan perangkat lunak

Dalam pembuatan alat tak luput dari program yang dijalankan agar alat bisa bekerja secara semestinya. Perangkat lunak digunakan sebagai jembatan untuk menjalankan sistem melalui eksekusi program. Untuk pemrograman digunakan perangkat lunak arduino IDE sebagai eksekusi program pada alat.

4.2.3 Kebutuhan Fungsional

Untuk melihat fungsi alat pengiriman suara secara nyata maka penulis melakukan penjabaran kebutuhan fungsional berikut:

1. perangkat bisa menginput suara secara realtime dengan menggunakan transmitter.
2. Perangkat dapat menghasilkan output berupa suara pada receiver.

4.2.4 Kebutuhan Non Fungsional

Terdapat beberapa Kebutuhan non fungsional dari alat pengiriman suara yaitu sebagai berikut:

1. Sistem bisa mengirim dan menerima data suara melalui modul nrf, bluetooth, dan wifi.
2. Sistem mampu menggunakan mikrokontroler Arduino Uno sebagai pengolah data.

4.3 Kebutuhan antarmuka

Kebutuhan antarmuka merupakan kebutuhan pokok dalam melakukan penelitian. Kebutuhan ini merupakan kebutuhan interaksi langsung antara alat pengiriman suara dengan pengguna terdapat beberapa Kebutuhan antarmuka yang terhubung dengan pengguna, perangkat keras maupun perangkat lunak.

4.3.1 Kebutuhan Antarmuka Pengguna

Dalam penggunaan alat ini tidak luput dari campur tangan penggunanya, perangkat ini didesain seminimal dan senyaman mungkin agar mudah digunakan. pengguna bisa menggunakan masukan dari mic yang ada pada transmitter sebagai masukan data pada perangkat.

4.3.2 Kebutuhan Antarmuka Perangkat Keras

Kebutuhan ini merupakan hubungan sesama perangkat keras yang digunakan, agar perangkat terlihat rapi dan kompleks maka dibuat kebutuhan antarmuka perangkat keras yang terbagi sebagai berikut:

1. Antarmuka modul dan mikrokontroler
Antarmuka modul dan mikro kontroler terdapat modul nrf, bluetooth, dan wifi yang terhubung dengan mikro kontroler Arduino uno.
2. Antarmuka mikro kontroler dan input data
Pada antarmuka mikro kontroler dan input data menghubungkan mikro kontroler Arduino uno dengan mic.

4.3.3 Kebutuhan Antarmuka Perangkat Lunak

Pada kebutuhan ini dilakukan program yang digunakan oleh transmitter dan receiver tiap module. Badan program dapat terhubung dengan sistem yang dijalankan menggunakan Arduino IDE. Arduino IDE adalah program perangkat lunak yang menggunakan pemrograman bahasa C.

Selain itu juga digunakan perangkat lunak sndpeek memiliki fungsi untuk output data yang bisa dilihat dalam data suara dan grafik dari suara yang dihasilkan. Program sndpeek merupakan program realtime audio visualization yang bisa menggunakan mic sebagai inputan suara.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

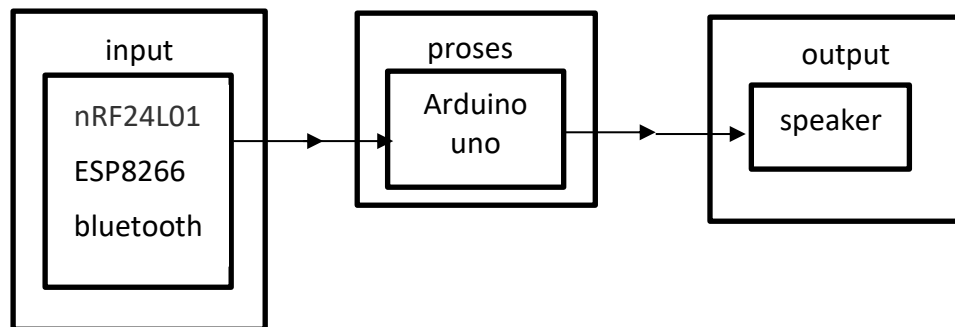
Pada bab ini penulis melakukan perancangan alat. Dalam melakukan perancangan penulis membuat skema rangkaian dan flowchar program. Selain itu juga dilakukan implementasi perangkat berdasarkan rancangan yang telah dibuat.

5.1 Perancangan Sistem

Pada perancangan sistem dijelaskan secara umum bagaimana sistem bekerja, juga dilakukan perancangan berdasarkan gambaran umum tersebut.

5.1.1 Gambaran Umum Sistem

Gambaran umum sistem menjelaskan tentang blok diagram serta hubungan nya dengan perangkat keras. Pada Gambar 5.1 akan menjelaskan hubungan proses input dan proses output pada alat.



Gambar 5.1 Diagram Blok Perancangan Sistem

Untuk modul input dan pengiriman data suara menggunakan tiga modul yaitu NRF24L01, ESP8266, dan HC-02. Arduino uno digunakan sebagai proses sistem, dan untuk output yang dihasilkan sistem menggunakan speaker.

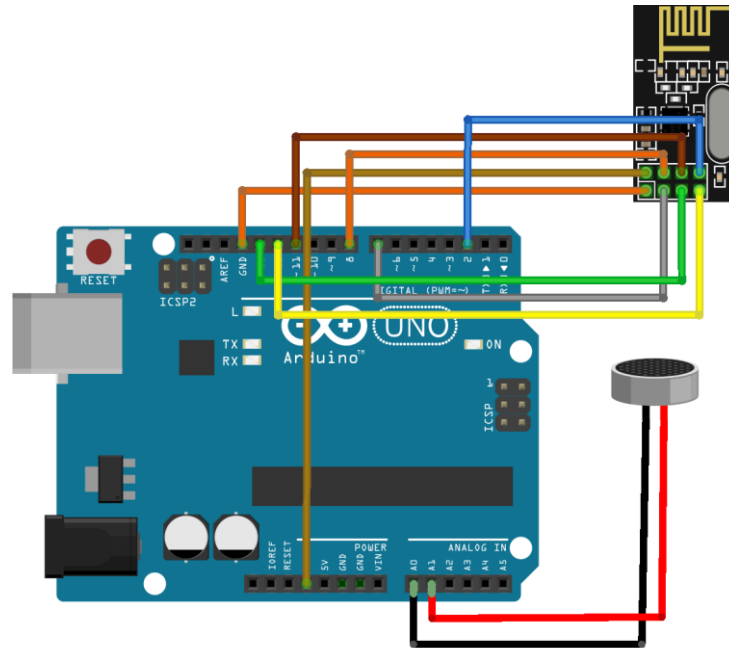
5.1.2 Perancangan Perangkat Keras

Pada bagian perancangan perangkat keras penulis membahas bagaimana perancangan perangkat keras secara mendetail dalam melakukan penelitian ini. Perancangan perangkat keras berupa perancangan modul pada transmitter dan receiver.

5.1.2.1 Perancangan modul pada transmitter

Pada sistem transmitter akan dilakukan perancangan modul yang digunakan untuk pengiriman data. Ketiga modul tersebut adalah NRF24L01, modul HC-05, dan modul ESP8266. Pada masing-masing transmitter, modul akan dipasang sesuai dengan pin yang digunakan pada papan board Arduino Uno.

Pada pengiriman suara melalui nrf menggunakan modul nRF24L01. Penjelasan dan rangkaian alat dilakukan secara mendetail pada gambar 5.2 berikut.



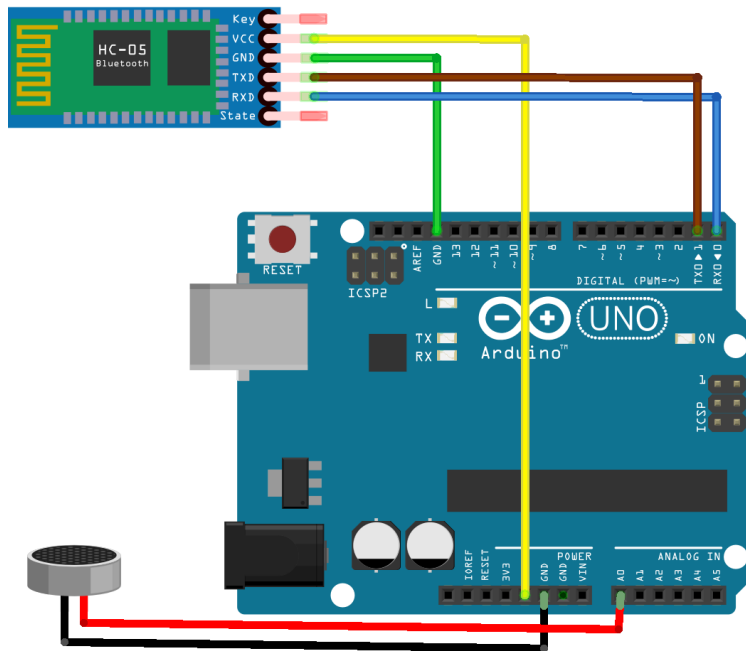
Gambar 5.2 Perancangan Transmitter pada NRF

Pada Gambar 5.2 bisa dilihat rancangan transmitter pada modul nRF24L01. Dalam rancangan tersebut menggunakan pin 7 dan 8 pada arduino sebagai radio transmit. Untuk mic menggunakan pin A0 sebagai input data analog.

Untuk letak pinout antara NRF24L01 dengan pemasangan board Arduino Uno dan mic dijabarkan dalam tabel 5.1.

Tabel 5.1 Hubungan Pinout nRF24L01 Pada board Arduino Uno

nRF24L01	Arduino uno	Mic
Pin ground	Pin ground	-
Pin vcc	Pin 3.3 v	-
Pin ce	Pin 7	-
Pin csn	Pin 8	-
Pin SCK	Pin 13	-
Pin MOSI	Pin 11	-
Pin MISO	Pin 12	-
Pin IRQ	Pin 2	-
-	Pin a0	Vcc
-	Pin a1	Ground



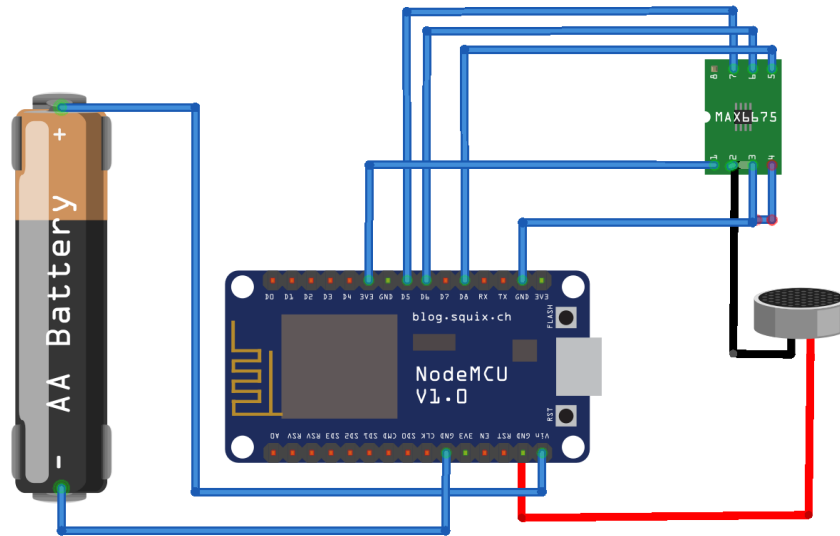
Gambar 5.3 Perancangan Transmitter pada Bluetooth

Gambar 5.3 merupakan rancangan transmitter pada modul bluetooth. Untuk serial komunikasi pada modul HC-05 menggunakan pinout rx. Pin tersebut terhubung dengan pin D0 dan pin tx yang dipasangkan pada pinout D1.

Untuk letak pinout antara modul HC-05 dengan pemasangan Arduino Uno dan mic dijabarkan dalam tabel 5.2.

Tabel 5.2 Hubungan Pinout HC-05 Pada Arduino Uno

Bluetooth	Arduino Uno	Mic
Pin vcc	Pin 5 v	-
Pin ground	Pin ground	-
Pin rx	Pin D0	-
Pin tx	Pin D1	-
-	Pin a0	Vcc
-	Ground	Ground



Gambar 5.4 Perancangan Transmitter pada WIFI

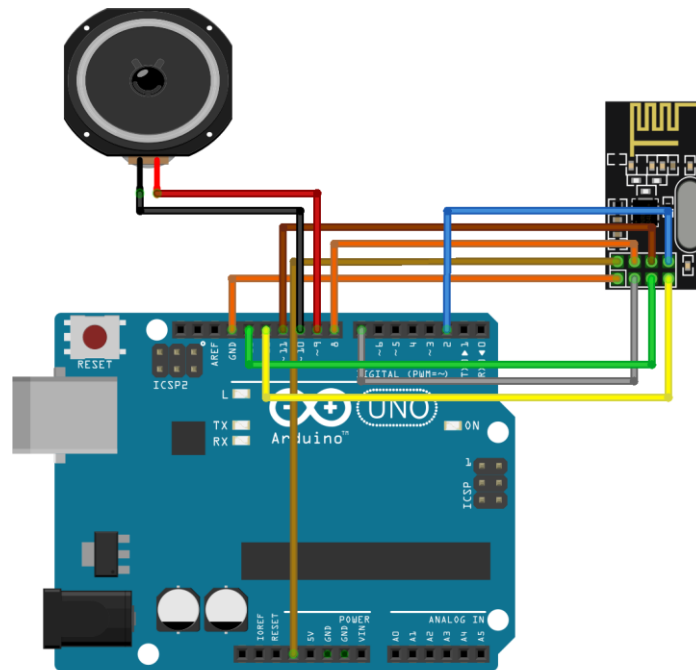
Gambar 5.4 menjabarkan bagaimana rancangan transmitter pada modul Esp 8266. Untuk fungsi ADC pada MCP3201 digunakan pin MISO yang dihubungkan dengan pin D6 sebagai slave line untuk mengirim data ke master. Pin CLK digunakan sebagai clock data pada data pin. Pin CS digunakan sebagai sirkuit terintegrasi yang dihubungkan dengan master.

Untuk letak pinout antara ESP8266 dengan Arduino Uno dapat dilihat pada tabel 5.3.

Tabel 5.3 Hubungan Pinout ESP 8266 pada Arduino Uno

Esp 8266	MCP3201
Pin 3v	Pin Vref
Pin Vin	Pin in+
Pin ground	Pin in-
Pin ground	Pin Vss
Pin D8	Pin CS/SHDN
Pin D6	Pin Dout/MISO
Pin D5	Pin CLK
Pin 3v	Pin Vdd

5.1.2.2 Perancangan modul reciever



Gambar 5.5 Perancangan Reciever pada NRF

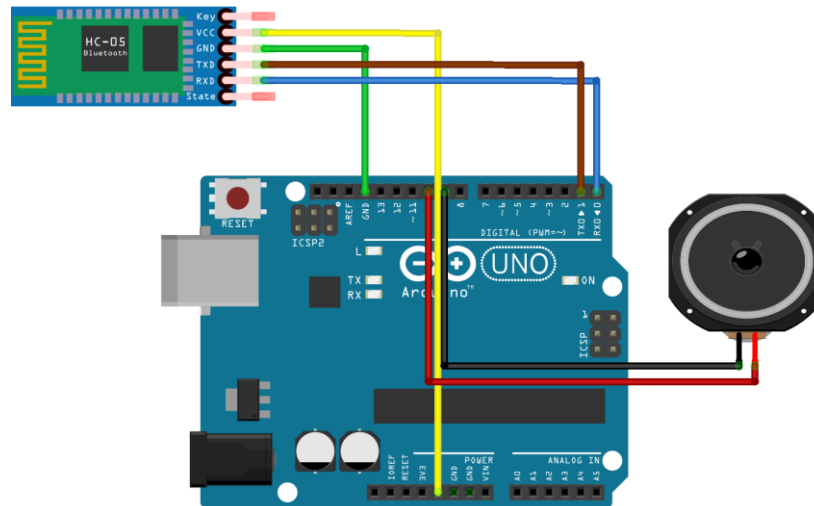
Pada Gambar 5.5 merupakan rancangan transmitter pada modul nRF24L01. Dalam rancangan tersebut menggunakan pin 7 dan 8 pada arduino sebagai radio transmit. Untuk mic menggunakan pin A0 sebagai input data analog

Untuk letak pinout antara NRF24L01 menggunakan Arduino Uno dan speaker dijabarkan secara jelas dalam tabel 5.4.

Tabel 5.4 Hubungan Pinout nRF24L01 pada Arduino Uno

nRF24L01	Arduino uno	Speaker
Pin ground	Pin ground	-
Pin vcc	Pin 3.3 v	-
Pin ce	Pin 7	-
Pin csn	Pin 8	-
Pin SCK	Pin 13	-
Pin MOSI	Pin 11	-

Pin MISO	Pin 12	-
Pin IRQ	Pin 2	-
-	Pin 9	Vcc
-	Pin 10	Ground



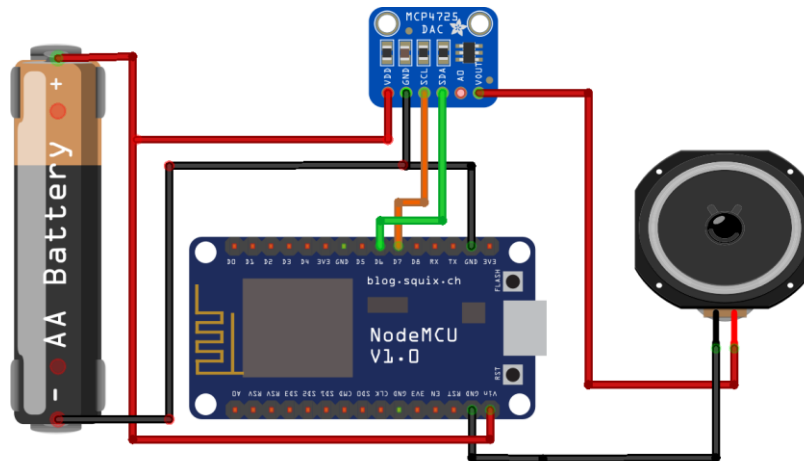
Gambar 5.6 Perancangan Reciever Pada Bluetooth

Gambar 5.6 menjelaskan perancangan transmiter pada modul bluetooth. Untuk serial komunikasi pada modul hc-05 digunakan pin rx. Pin tersebut terhubung dengan pinout D0 dan pinout tx yang dihubungkan dengan pinout D1.

Untuk letak pinout antara HC-05 menggunakan Arduino Uno dan speaker dijabarkan secara jelas dalam tabel 5.4.

Tabel 5.5 Hubungan Pinout HC-05 pada Arduino Uno

Bluetooth	Arduino Uno	Speaker
Pin vcc	Pin 5 v	-
Pin ground	Pin ground	-
Pin rx	Pin D0	-
Pin tx	Pin D1	-
-	Pin 9	Vcc
-	Pin 10	Ground



Gambar 5.7 Rancangan Reciever sistem WIFI

Gambar 5.7 merupakan rancangan reciever pada modul Esp 8266. Untuk fungsi ADC pada MCP3201 digunakan pin MISO yang dihubungkan dengan pin D6 sebagai slave line untuk mengirim data ke master. Pin CLK digunakan sebagai clock data pada data pin. Pin CS digunakan sebagai sirkuit terintegrasi yang dihubungkan dengan master.

Untuk letak pinout antara ESP8266 dengan MCP42725 dan mic dapat dilihat secara jelas didalam tabel 5.6.

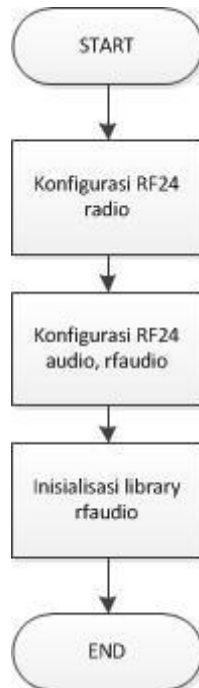
Tabel 5.6 Hubungan Pin ESP 8266 pada MCP4725

ESP8266	MCP4725
Pin vin	Pin vdd
Pin ground	Pin ground
Pin d7	Pin scl
Pin d6	Pin sda

5.1.3 Perancangan Perangkat Lunak

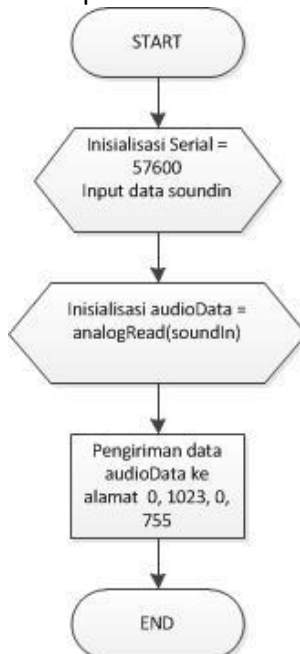
Pada bagian ini dijelaskan bagaimana cakra kerja alur sistem dalam bentuk program. Program tersebut digunakan pada mikrokontroler dan dijabarkan kedalam bentuk flowchart.

Pada Gambar 5.8 dapat dilihat alur sistem pada nRf. Proses dimulai dari konfigurasi RF24 radio, selanjutnya konfigurasi RF24 audio dan rfaudio. Setelah konfigurasi kemudian dilakukan inisialisasi library rfaudio untuk memproses suara yang diterima oleh arduino Uno. Setelah data diproses maka transmitter akan memproses hasil yang didapatkan kemudian dikirimkan ke reciever.



Gambar 5.8 Perancangan Kalibrasi modul nrf

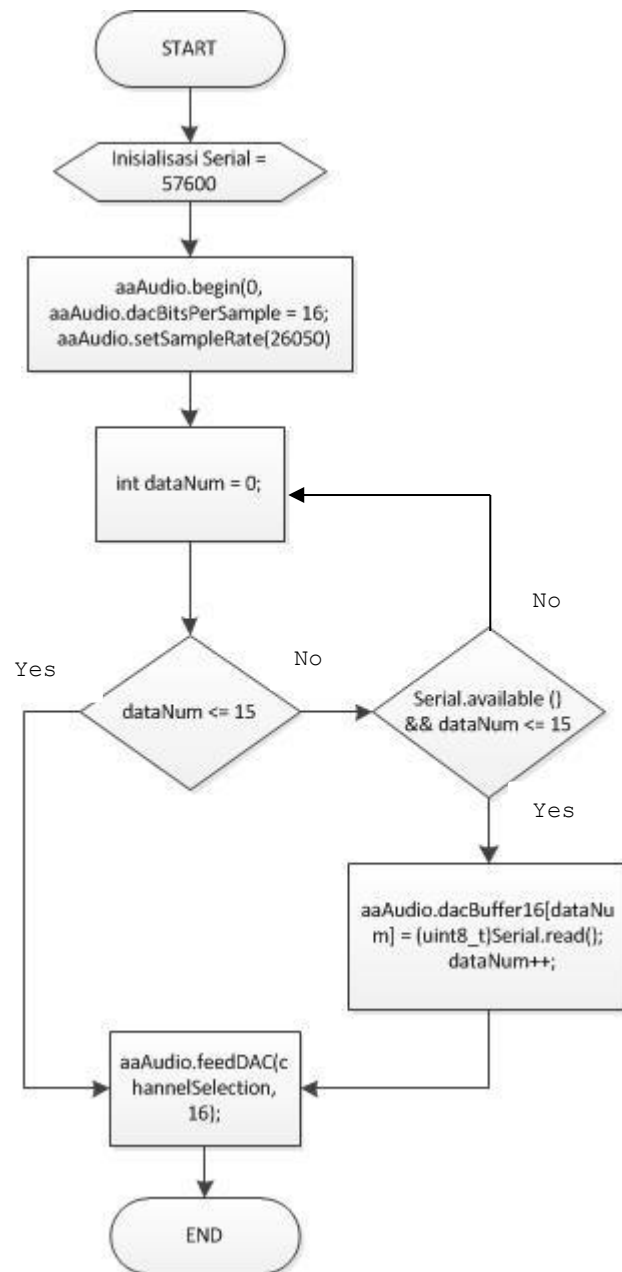
Gambar 5.9 merupakan alur sistem pada modul bluetooth. Untuk serial komunikasi pada transmitter dan receiver menggunakan inisialisasi serial 57600. Setelah kedua perangkat terhubung maka dilakukan inisialisasi audiodata. Kemudian dilakukan pengiriman data pada alamat bluetooth yang ditentukan.



Gambar 5.9 Perancangan Kalibrasi modul bluetooth

Gambar 5.10 merupakan alur sistem pada wifi. Untuk serial komunikasi pada transmitter dan receiver menggunakan inisialisasi serial 57600. Setelah kedua perangkat terhubung maka dilakukan pembacaan data dengan sampelbit 16 dan

sampelrate(26050). Jika datanum lebih atau sama dengan 15 maka akan dilakukan pengolahan data. Jika kurang dari 15 maka akan mencari serial yang tersedia dan melakukan buffer suara yang kemudian akan dilakukan pengolahan data.



Gambar 5.10 Perancangan Kalibrasi modul wifi

5.2 Implementasi

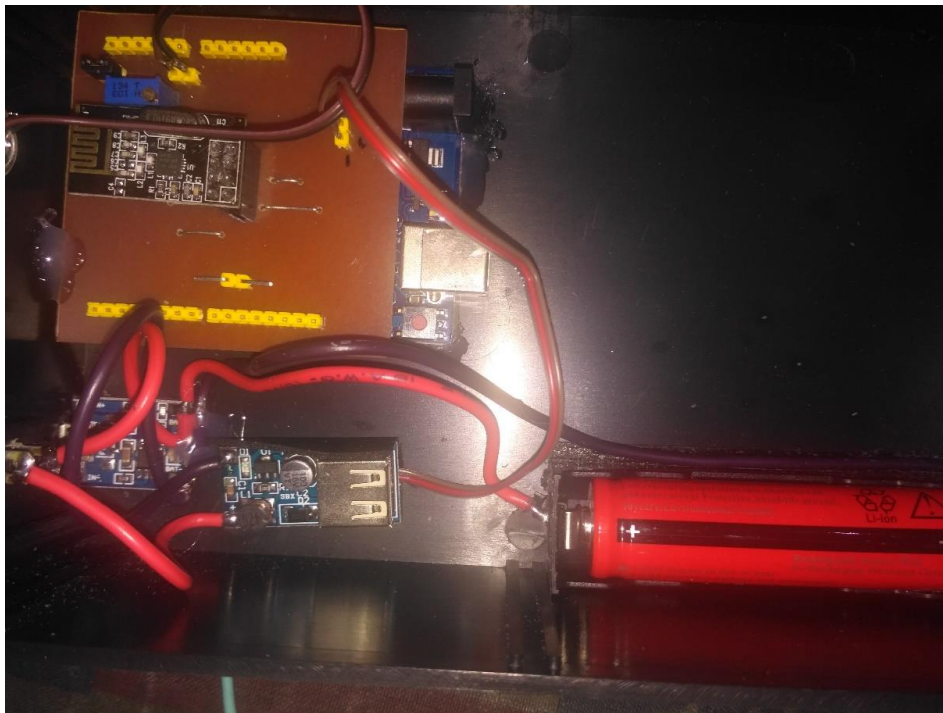
Implementasi merupakan sub bab yang berisi bagaimana perakitan perangkat pada sistem. Perakitan ini dapat berupa berupa bentuk fisik perangkat maupun program yang dijalankan. Implementasi sistem akan dijabarkan lebih lanjut kedalam penjelasan berikut.

5.2.1 Implementasi Perangkat Keras

Setelah melakukan perancangan, tahap selanjutnya adalah melakukan implementasi. Dalam tahap implementasi perangkat keras, bagian yang disambungkan disesuaikan dengan letak pinout pada rancangan. Terdapat dua implementasi perangkat keras yang akan dijelaskan yaitu implementasi modul transmiter dan implementasi modul receiver.

5.2.1.1 Implementasi modul transmiter

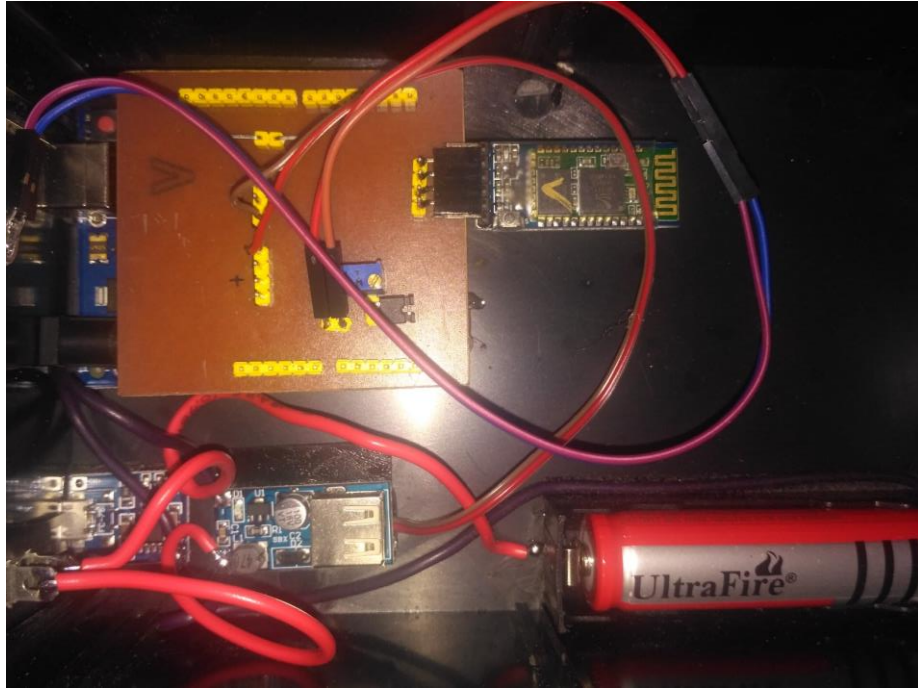
Pada gambar 5,11 diperlihatkan bagaimana melakukan penempatan pada modul dan mikrokontroler, penempatan ini dibuat berdasarkan perancangan modul pada transmiter. Modul pengiriman akan terhubung dengan arduino Uno yang sudah diberi daya baterai dan terhubung dengan mic.



Gambar 5.11 Implementasi transmiter nrf

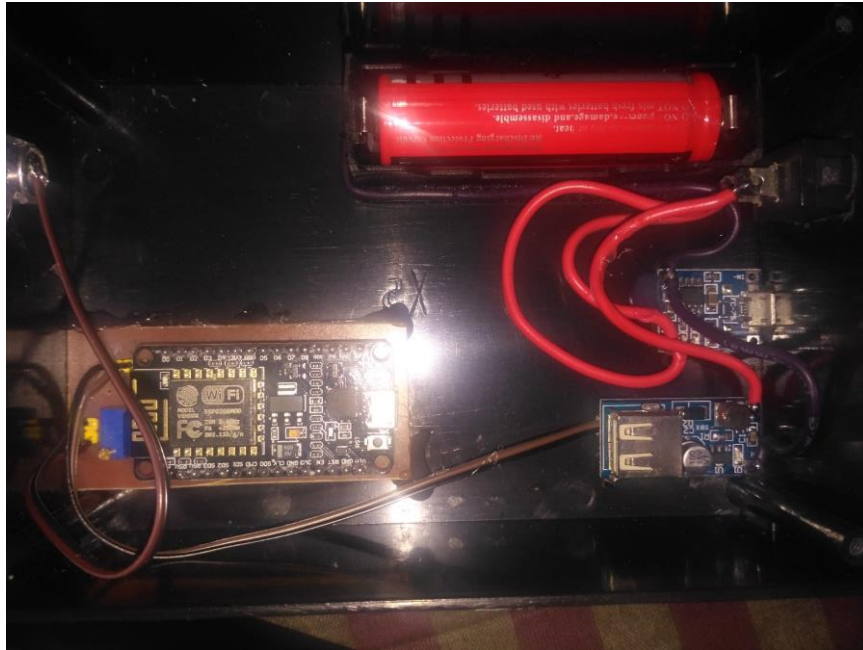
Pada gambar 5.12 dilakukan penempatan sensor, penempatan ini disesuaikan dengan perancangan modul pada transmiter. Modul pengiriman

akan terhubung dengan arduino Uno yang sudah diberi daya baterai dan terhubung dengan mic.



Gambar 5.12 Implementasi transponder bluetooth

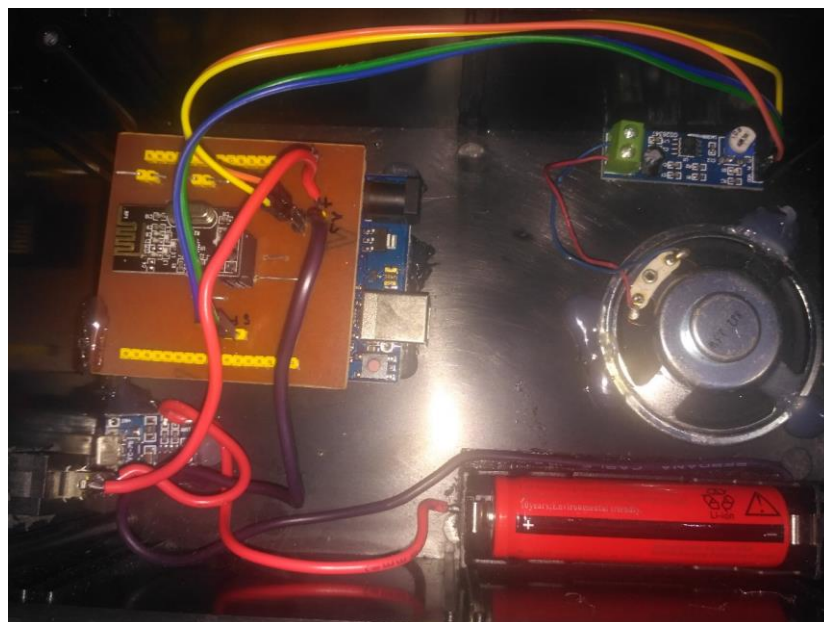
Gambar 5.13 merupakan implementasi transponder wifi. Modul ESP8266 yang sudah terpasang dengan MCP4725 dihubungkan dengan mic sebagai inputan suara dan baterai sebagai daya.



Gambar 5.13 Implementasi transmitter wifi

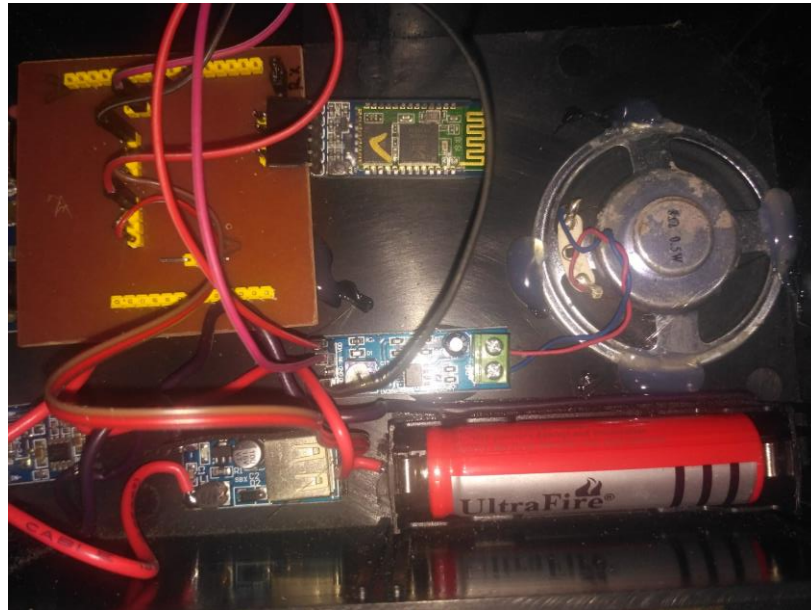
5.2.1.2 Implementasi modul reciever

Pada bagian ini modul dan mikrokontroler dipasang dengan bagian output pada alat. Penempatan perangkat disesuaikan dengan perancangan modul pada reciever. Modul pengiriman akan terhubung dengan arduino Uno yang sudah diberi daya baterai dan dihubungkan dengan speaker seperti yang terlihat pada gambar 5.14.

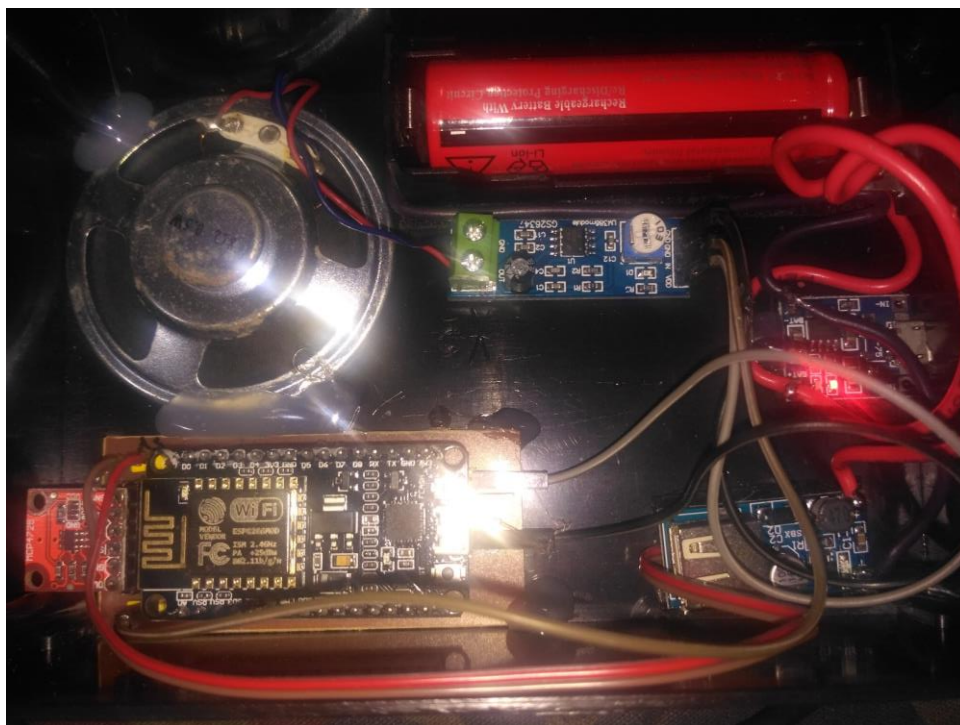


Gambar 5.14 Implementasi reciever nrf

Gambar 5.15 merupakan implementasi receiver bluetooth. Modul ESP8266 yang sudah terpasang dengan arduino uno dihubungkan dengan speaker sebagai output suara dan baterai sebagai daya.



Gambar 5.15 Implementasi receiver bluetooth



Gambar 5.16 Implementasi receiver wifi

Gambar 5.16 merupakan implementasi receiver wifi. Modul ESP8266 yang sudah terpasang dengan MCP4725 dihubungkan dengan speaker sebagai output suara dan baterai sebagai daya.

5.2.2 Implementasi Perangkat Lunak

Pada tahap ini dilakukan implementasi program pada alat yang sudah dibuat. Dalam melakukan implementasi program berdasarkan flowchart dan alur yang telah dibuat sebelumnya, kemudian disusun kedalam sebuah program untuk menjalankan perangkat keras.

5.2.2.1 Implementasi program nrf

Pada bagian ini merupakan program inisialisasi pada modul nrf. Pada Tabel 5.7 adalah program untuk modul NRF. Program ini menggunakan library RF24Audio yang terdapat pada aplikasi Arduino Uno. Untuk pin menggunakan pin 7 dan 8. Dan untuk inisialisasi RF24audio menggunakan pin 1.

Tabel 5.7 Program nrf

Program MPU6050	
1	#include <RF24.h>
2	#include <SPI.h>
3	#include <RF24Audio.h>
4	
5	RF24 radio(7,8);
6	RF24Audio rfAudio(radio,1);
7	
8	void setup() {
9	rfAudio.begin();
10	}

5.2.2.2 Implementasi program bluetooth

Pada Tabel 5.8 program bluetooth menggunakan baudrate 57600, untuk pinmode menggunakan pinmode soundin yang sebelumnya telah diinisialisasi. Selanjutnya pembacaan analog data pada pinmode soundin, dan write data pada alamat bluetooth yang telah ditentukan.

Tabel 5.8 Program transmit bluetooth

Program Transmit Bluetooth	
1	float soundIn = A1;
2	void setup()
3	{
4	Serial.begin(57600);
5	pinMode(soundIn, INPUT);
6	}

7	void loop()
8	{
9	uint16_t audioData = analogRead(soundIn);
10	Serial.write(map(audioData, 0, 1023, 0,
11	755));
	}

Pada Tabel 5.9 program reciever bluetooth dilakukan inisialisasi datanum, jika datanum kurang dari sama dengan 15 maka akan dilakukan penncarian serial yang tersedia. Kemudian dilakukan buffer data yang berupa suara.

Tabel 5.9 Program reciever bluetooth

Program Reciever Bluetooth	
1	uint32_t dispTimer = 0;
2	uint8_t channelSelection = 0;
3	void loop()
4	{
5	RX();
6	}
7	uint32_t dynSampleRate = 0;
8	void RX()
9	{
10	int dataNum = 0;
11	while (dataNum <= 15)
12	{
13	while (Serial.available () && dataNum <=
14	15)
15	{
16	aaAudio.dacBuffer16[dataNum] =
17	(uint8_t)Serial.read();
18	dataNum++;
19	}
20	}
21	aaAudio.feedDAC(channelSelection, 16);
22	}

5.2.2.3 Implementasi program wifi

Pada Tabel 5.10 implementasi program wifi dilakukan komunikasi agar transmitter dan reciever saling terhubung dengan node router wifi. Untuk komunikasi dengan router maka transmitter dan reciever saling dicocokkan dengan username dan password yang digunakan router.

Tabel 5.10 Program komunikasi antarwifi

Program Komunikasi Wifi	
1	WiFi.setOutputPower(10); // reduce power to 10dBm = 10mW
2	WiFi.mode(WIFI_STA);
3	WiFi.begin("Redmi3X", "12345");
4	
5	Serial.println("");
6	Serial.println("Connecting to wifi");
7	while (WiFi.status() != WL_CONNECTED) {
8	delay (500);
9	Serial.print (".");
10	}
11	
12	Serial.println ("");
13	Serial.print ("Cnnectd to ");
14	Serial.println ("g1");
15	Serial.print ("IP ");
16	Serial.println (WiFi.localIP());

Pada Tabel 5.11 dilakukan implementasi program transmit wifi. Readptr berfungsi untuk inialisasi sinyal adc. Jika filter highpass berjalan maka fungsi readptr akan berjalan dengan inialisasi filterloop(val)+2048.

Tabel 5.11 Program transmit wifi

Program Transmit Wifi	
1	readptr = &adc_buf[!current_adc_buf][i];
2	int32_t val = *readptr;
3	int32_t rectified;
4	
5	if (enable_highpass_filter)
6	{
7	*readptr = filterloop(val) + 2048;
8	val = *readptr;
9	}
10	
11	rectified = abs(val - silence_value);
12	
13	accum_silence += val;

14	envelope_value += rectified;
15	writeptr = delta7_sample(last, readptr, writeptr);
16	last = val;

Pada Tabel 5.12 dilakukan implementasi program reciever wifi dilakukan inisialisasi untuk penyimpanan cache ram dan audio. Jika tida terjadi penerimaan data, maka akan menunggu data dari transmit. Dan melakukan filter DAC, dari data yang diterima.

Tabel 5.12 Program reciever wifi

Program Reciever Wifi	
1	void ICACHE_RAM_ATTR playsample_isr(void)
2	{
3	if (play_waiting)
4	{
5	return;
6	}
7	DAC(data_buf[current_play_data_buf][play_data_buf_pos]);
8	play_data_buf_pos++;
9	
10	if (play_data_buf_pos >= sizeof(data_buf[0]) /
11	sizeof(data_buf[0][0]))
12	{
13	play_data_buf_pos = 0;
14	current_play_data_buf++;
15	
16	if (current_play_data_buf == NB_DATA_BUFS)
17	{
18	current_play_data_buf = 0;
19	}
20	
21	if (current_play_data_buf == current_rcv_data_buf)
22	{
23	play_waiting = true;
24	play_waiting_at = micros();
25	}
26	}
27	}

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab pengujian serta analisis sistem penulis mengerjakan uji coba alat jadi. Kemudian membuat analisa terhadap percobaan alat. Bab ini bertujuan untuk memperoleh data hasil dan analisis untuk menentukan bagaimana dan sistem manakah yang mempunyai performa lebih baik dalam pengiriman sinyal suara. Pengujian ini meliputi pengujian pada nrf, pengujian pada bluetooth, pengujian pada wifi, dan pengujian secara keseluruhan.

6.1 Pengujian Pada nRF24L01

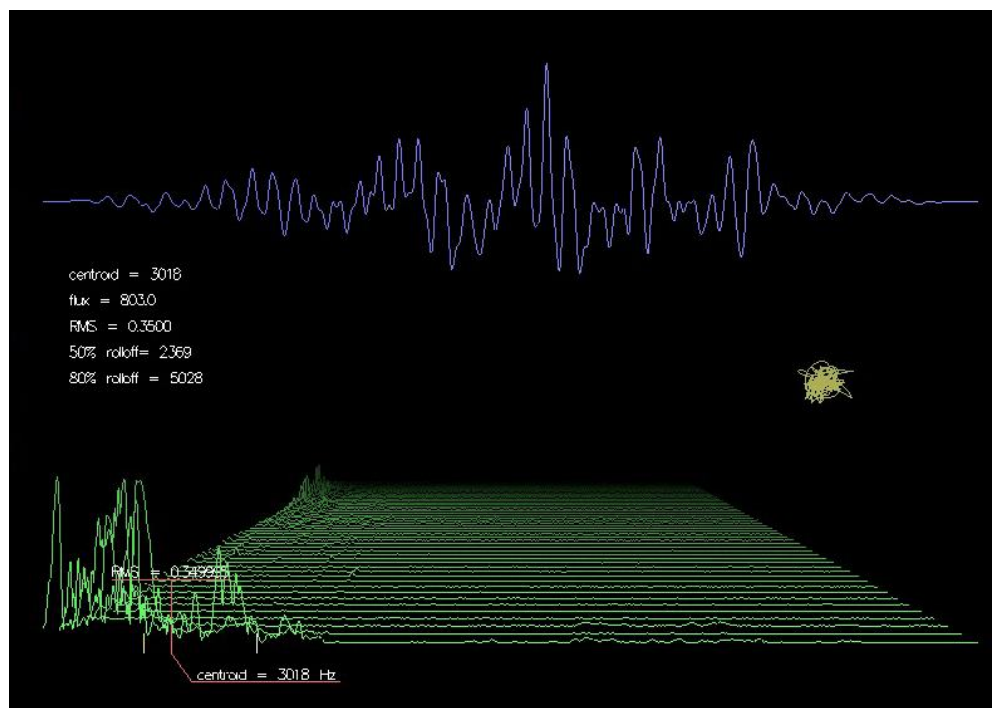
Pada tahap ini dilakukan pengujian dengan menggunakan modul nRF24L01. untuk pengujian sendiri dilakukan dalam tiga tahap yang merupakan tujuan dilakukannya pengujian, Prosedur Pengujian, serta Hasil dan Analisa.

6.1.1 Tujuan

Pengujian modul nRF24L01 bertujuan untuk menentukan nilai suara yang dihasilkan. Dari nilai yang didapatkan kemudian dilakukan pencatatan data untuk kemudian dilakukan analisis data.

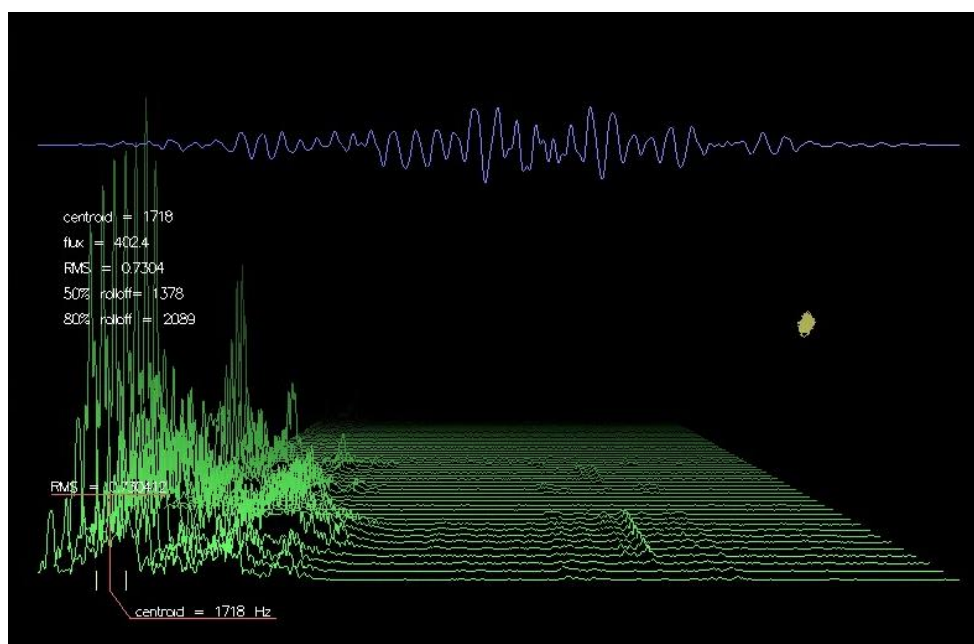
6.1.2 Prosedur Pengujian

Cara uji coba modul NRF24L01 menggunakan 3 tahap yaitu pengujian dengan selisih jarak antar node 1 meter, 3 meter, dan 6 meter. Hal ini dilakukan untuk mengukur jarak yang bisa dihasilkan menggunakan modul nRF24L01. Dalam pengujian ini dilakukan dengan mengucapkan kata “tes” sebagai inputan suara.



Gambar 6.1 Percobaan nrf dengan selisih 1 meter

Pada Gambar 6.1 dapat dilihat percobaan alat dengan selisih antarnode 1 meter. Pengujian ini dilakukan berdasarkan jarak transmiter dan receiver yang berjarak 1 meter. Pengujian dilakukan dengan melihat data Centroid, RMS, dan flux. Pada gambar terdapat 3 grafik yang ditunjukkan. Grafik biru merupakan gambaran frekuensi yang dihasilkan. Pada kurva biru terlihat bahwa jika nilai gain tidak terlalu besar maka band frekuensi akan melebar. Tetapi pada nilai gain yang besar maka lebar band frekuensi menjadi sempit. Oleh karena itu hasil respon keseluruhan dari equaliser grafik akan terbentuk gelombang kecil-kecil memanjang dan bukannya bentuk grafik respon audio yang mulus. Grafik hijau merupakan RMS yang dihasilkan dan grafik kuning adalah rolloff yang merupakan frekuensi audio di sepanjang sumbu horizontal dan output di sepanjang vertikal dalam desibel (db).

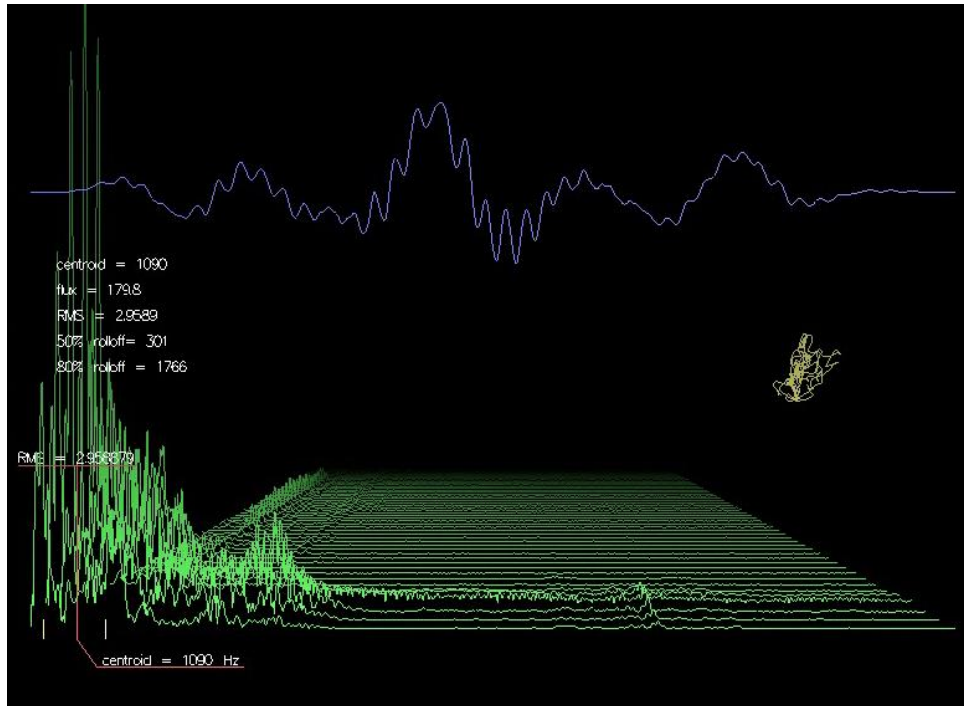


Gambar 6.2 Percobaan nrf dengan selisih 3 meter

Pada Gambar 6.2 dapat dilihat pengujian pada jarak 3 meter. Pada gambar terlihat penurunan secara signifikan. Pada gambar terdapat 3 grafik yang ditunjukkan. Grafik biru merupakan gambaran frekuensi yang dihasilkan. Pada kurva biru terlihat bahwa jika nilai gain tidak terlalu besar maka band frekuensi akan melebar. Tetapi pada nilai gain yang besar maka lebar band frekuensi menjadi sempit. Oleh karena itu hasil respon keseluruhan dari equaliser grafik akan terbentuk gelombang kecil yang beraturan. Pada grafik terlihat perbedaan frekuensi dengan pengujian sebelumnya. Grafik hijau merupakan RMS yang dihasilkan berupa 0,7304. Grafik kuning adalah rolloff yang merupakan frekuensi audio di sepanjang sumbu horizontal dan output di sepanjang vertikal dalam desibel (db) dan didapatkan roll off dalam rentang 1378 sampai dengan 2089.

Pada Gambar 6.3 dapat dilihat pengujian modul nrf berjarak pada antarnode 6 meter. Dalam gambar terjadinya perubahan frekuensi yang dihasilkan. Pada kurva biru terlihat bahwa jika nilai gain tidak terlalu besar maka band frekuensi

akan melebar. Tetapi pada nilai gain yang besar maka lebar band frekuensi menjadi sempit. Oleh karena itu hasil respon keseluruhan dari equaliser grafik akan terbentuk gelombang kecil berupa yang memanjang. Pada gambar terdapat 3 grafik yang ditunjukkan. Grafik biru merupakan gambaran frekuensi yang dihasilkan.pada grafik terlihat frekuensi berupa 1090 hz. Grafik hijau merupakan RMS yang dihasilkan berupa 2,9589. Grafik kuning adalah rolloff yang merupakan frekuensi audio di sepanjang sumbu horisontal dan output di sepanjang vertikal dalam desibel (db) dan didapatkan roll off dalam rentang 304 sampai dengan 1766.



Gambar 6.3 Percobaan nrf dengan selisih 6 meter

6.1.2 Hasil dan Analisis

Hasil dan analisa didapatkan setelah melakukan uji coba alat, kemudian mencatat data yang didapatkan. Hasil data yang didapatkan setelah melakukan uji coba modul NRF24L01 di jabarkan pada tabel 6.1.

Tabel 6.1 Hasil Pengujian modul nrf

	Jarak		
Nama Timbral	1 meter	3 meter	6 meter
Centroid	3018 Hz	1718 Hz	1090 Hz
Flux	803,0	402,4	179,8
RMS	0.3500	0,7304	2,9589

Pada tabel 6.1 Nilai yang digunakan adalah nilai Timbral. Trimbe digunakan untuk mengukur karakteristik suara berdasarkan kualitasnya. Semakin besar nilai Centroid maka semakin jelas suara yang bisa kita dengar. Dari tabel dapat disimpulkan semakin jauh jarak transmiter dengan receiver maka suara yang dapat didengar semakin kurang jelas.

6.2 Pengujian Bluetooth

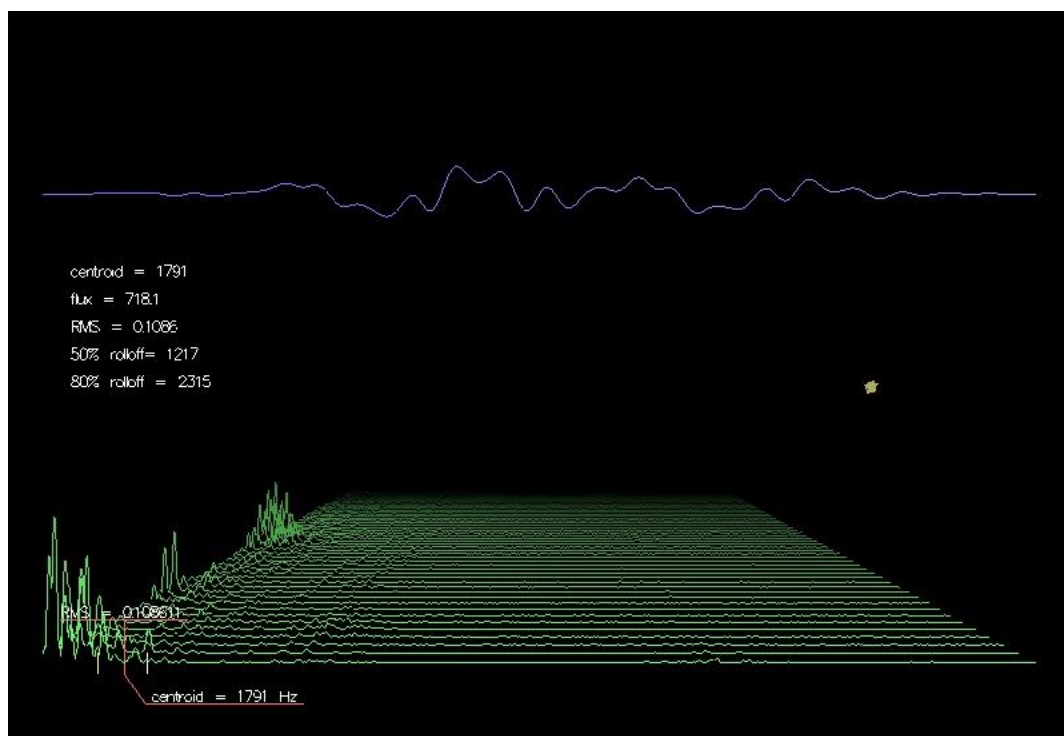
Pada tahap ini dilakukan pengujian dengan menggunakan modul HC-05 untuk pengujian sendiri dilakukan dalam tiga tahap yang merupakan tujuan dilakukannya pengujian, Prosedur Pengujian, serta Hasil dan Analisa.

6.2.1 Tujuan

Pengujian modul bluetooth bertujuan untuk menentukan nilai suara yang dihasilkan. Nilai ini yang kemudian digunakan sebagai acuan dalam melakukan analisis.

6.2.2 Prosedur

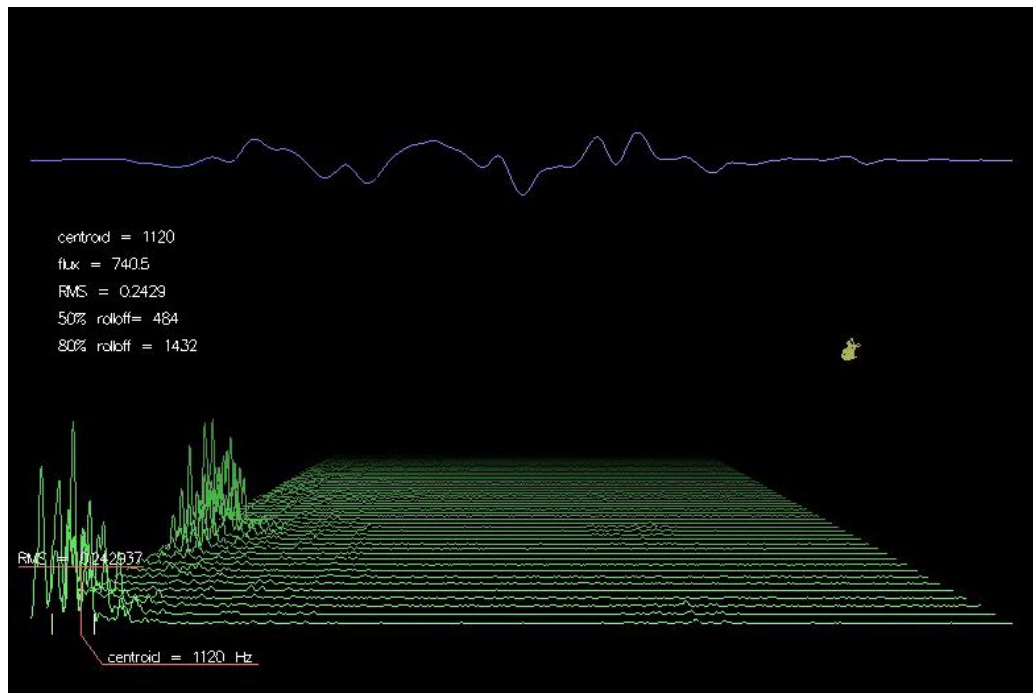
Cara uji coba modul HC-05 seperti percobaan sebelumnya yaitu menggunakan 3 tahap yaitu pengujian dengan selisih jarak antar node 1 meter, 3 meter, serta jarak antarnode 6 meter. Dalam pengujian ini dilakukan dengan mengucapkan kata “tes” sebagai inputan suara.



Gambar 6.4 Percobaan modul HC-05 dengan selisih 1 meter

Pada gambar 6.4 dilakukan ujicoba modul HC-05 dengan menggunakan jarak 1 meter. Hal ini didasari dari jarak kirim transmiter kepada receiver. Grafik biru

merupakan gambaran frekuensi yang dihasilkan. Pada kurva biru terlihat bahwa jika nilai gain tidak terlalu besar maka band frekuensi akan melebar. Tetapi pada nilai gain yang besar maka lebar band frekuensi menjadi sempit. Oleh karena itu hasil respon keseluruhan dari equaliser grafik akan terbentuk gelombang kecil-kecil memanjang dan bukannya bentuk grafik respon audio yang mulus. Pada grafik terlihat frekuensi berupa 1791 hz. Grafik hijau merupakan RMS yang dihasilkan berupa 0,1066. Grafik kuning adalah rolloff yang merupakan frekuensi audio di sepanjang sumbu horisontal dan output di sepanjang vertikal dalam desibel (db) dan didapatkan roll off dalam rentang 1217 sampai dengan 2315.

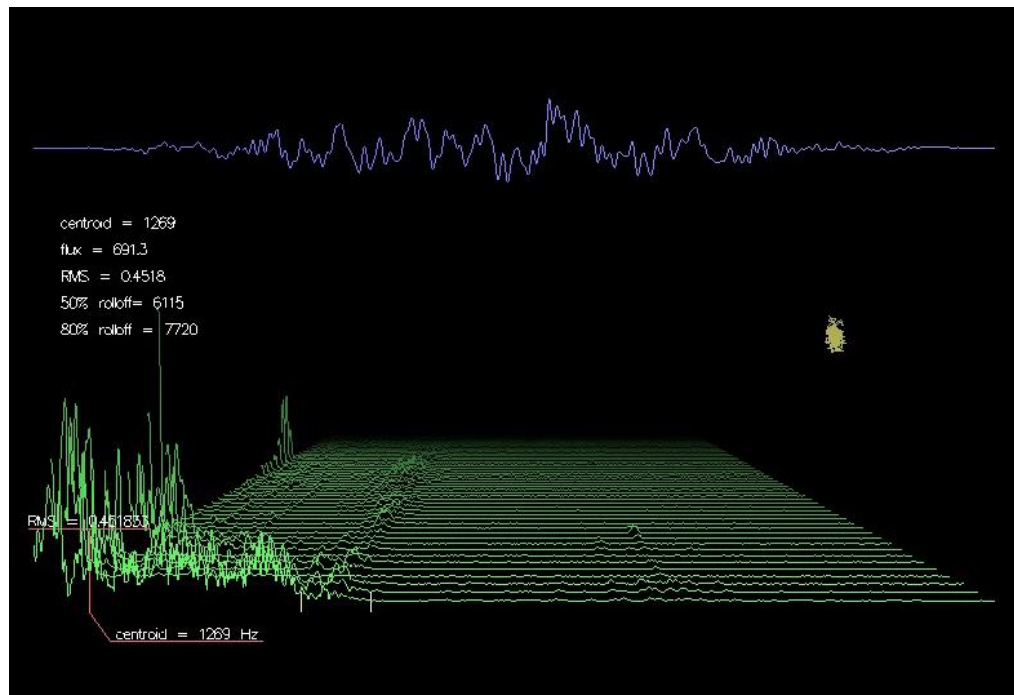


Gambar 6.5 Percobaan modul HC-05 dengan selisih 3 meter

Gambar 6.5 merupakan pengujian modul HC-05 pada jarak 3 meter. Grafik biru merupakan gambaran frekuensi yang dihasilkan. Pada kurva biru terlihat bahwa jika nilai gain tidak terlalu besar maka band frekuensi akan melebar. Tetapi pada nilai gain yang besar maka lebar band frekuensi menjadi sempit. Oleh karena itu hasil respon keseluruhan dari equaliser grafik akan terbentuk gelombang kecil-kecil memanjang dan bukannya bentuk grafik respon audio yang mulus. Pada grafik terlihat frekuensi berupa 1120 hz. Grafik hijau merupakan RMS yang dihasilkan berupa 0,2429. Grafik kuning adalah rolloff yang merupakan frekuensi audio di sepanjang sumbu horisontal dan output di sepanjang vertikal dalam desibel (db) dan didapatkan roll off dalam rentang 484 sampai dengan 1432.

Pada gambar 6.6 dilakukan ujicoba modul HC-05 dengan menggunakan jarak 6 meter. Data diambil berdasarkan jarak transmiter dan receiver. Grafik biru merupakan gambaran frekuensi yang dihasilkan. Pada kurva biru terlihat bahwa jika nilai gain tidak terlalu besar maka band frekuensi akan melebar. Tetapi pada nilai gain yang besar maka lebar band frekuensi menjadi sempit. Oleh karena itu

hasil respon keseluruhan dari equaliser grafik akan terbentuk gelombang kecil-kecil memanjang atau riak. Pada grafik terlihat frekuensi berupa 1269 hz. Grafik hijau merupakan RMS yang dihasilkan berupa 0,4518. Grafik kuning adalah rolloff yang merupakan frekuensi audio di sepanjang sumbu horisontal dan output di sepanjang vertikal dalam desibel (db) dan didapatkan roll off dalam rentang 6115 sampai dengan 7720.



Gambar 6.6 Percobaan modul HC-05 dengan selisih 6 meter

6.2.3 Hasil dan Analisis

Hasil dan analisa didapatkan setelah melakukan uji coba alat, kemudian mencatat data yang didapatkan. Setelah melakukan ujicoba modul HC-05, hasil yang didapat kemudian di jabarkan kedalam tabel 6.2.

Tabel 6.2 Hasil Pengujian modul Bluetooth

	Jarak		
Nama Timbral	1 meter	3 meter	6 meter
Centroid	1791 Hz	1120 Hz	1269 Hz
Flux	718,1	740,5	691,3
RMS	0.1086	0,2429	0,4518

Dari hasil yang didapatkan setelah ujicoba modul HC-05 kemudian disimpulkan data yang diterima secara signifikan tidak terlalu jauh meskipun

dilakukan dengan jarak yang berbeda. Tetapi kualitas suara masih terbilang tidak terlalu bagus.

6.3 Pengujian WIFI

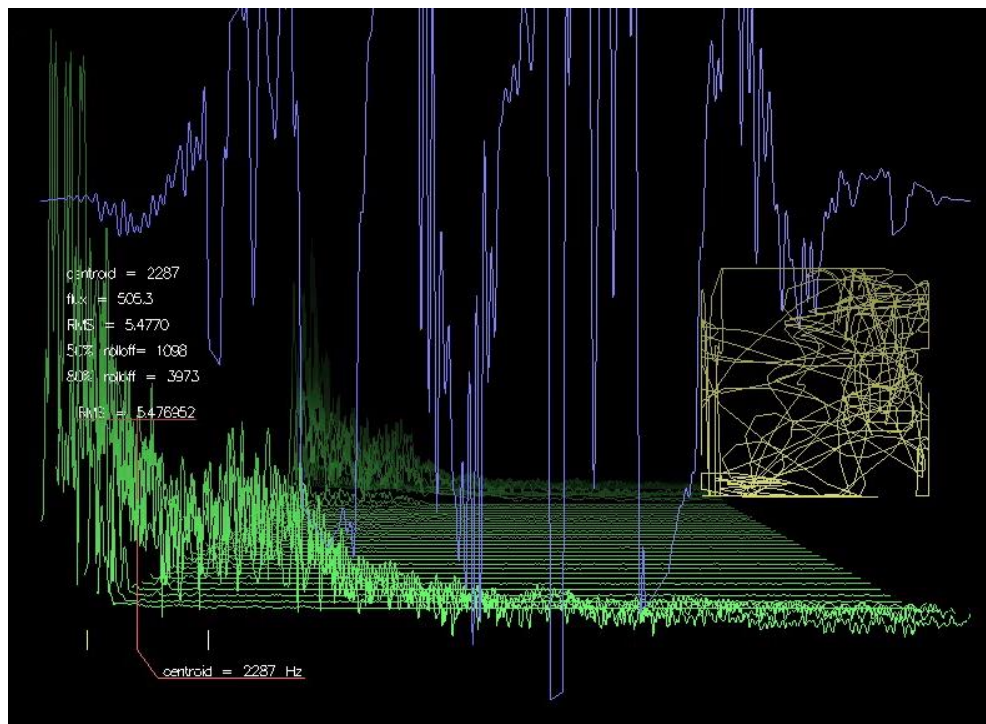
Pada tahap ini dilakukan pengujian dengan menggunakan modul ESP8266. Untuk pengujian sendiri dilakukan dalam tiga tahap yang merupakan tujuan dilakukannya pengujian, Prosedur Pengujian, serta Hasil dan Analisa.

6.3.1 Tujuan

Pengujian modul ESP8266 bertujuan untuk menentukan nilai suara yang dihasilkan. Nilai ini yang kemudian digunakan sebagai acuan dalam melakukan analisis.

6.3.2 Prosedur

Cara uji coba modul ESP8266 seperti percobaan sebelumnya yaitu menggunakan 3 tahap yaitu pengujian dengan selisih jarak antar node 1 meter, 3 meter, serta jarak antarnode 6 meter. Dalam pengujian ini dilakukan dengan mengucapkan kata “tes” sebagai inputan suara

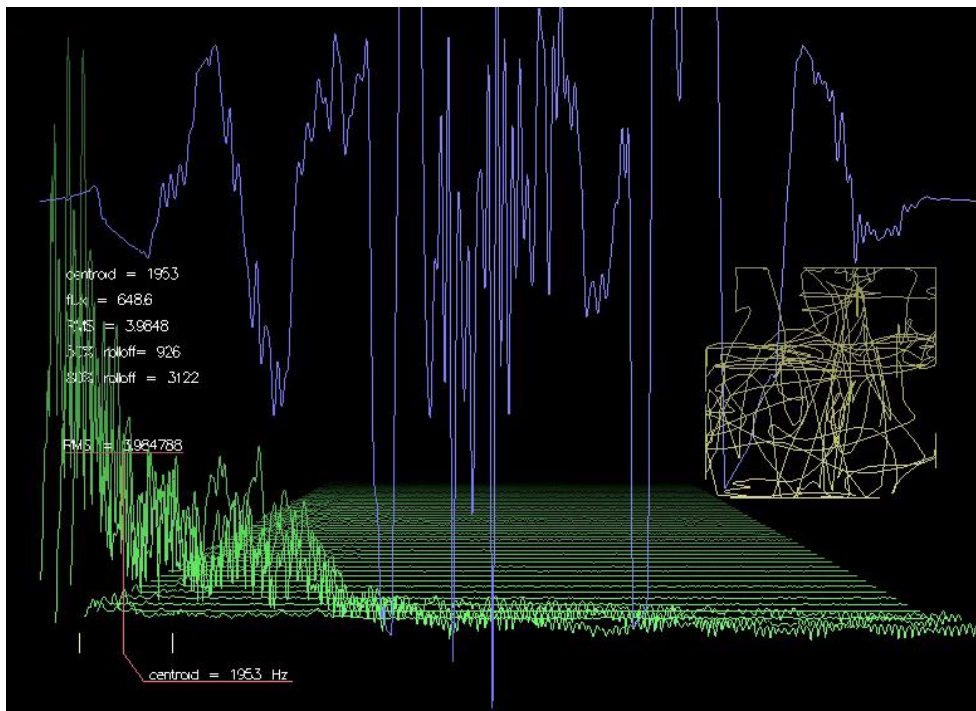


Gambar 6.7 Percobaan modul ESP8266 dengan selisih 1 meter

Pada Gambar 6.7 merupakan uji coba modul ESP8266 yang dilakukan. Uji coba ini dilakukan dengan jarak antara transmitter dan receiver sekitar 1 meter. Grafik biru merupakan gambaran frekuensi yang dihasilkan terdapat kenaikan frekuensi yang besar. Pada kurva biru terlihat bahwa jika nilai gain yang besar maka lebar band frekuensi akan besar sehingga jarak sampingnya mengecil. Tetapi pada nilai gain yang besar maka lebar band frekuensi menjadi sempit. Oleh karena itu hasil

respon keseluruhan dari equaliser grafik akan terbentuk gelombang besar yang memanjang dan melebar. Pada grafik terlihat frekuensi berupa 2287 hz. Grafik hijau merupakan RMS yang dihasilkan berupa 5,4477. Grafik kuning adalah rolloff yang merupakan frekuensi audio di sepanjang sumbu horisontal dan output di sepanjang vertikal dalam desibel (db) dan didapatkan roll off dalam rentang 1098 sampai dengan 3973.

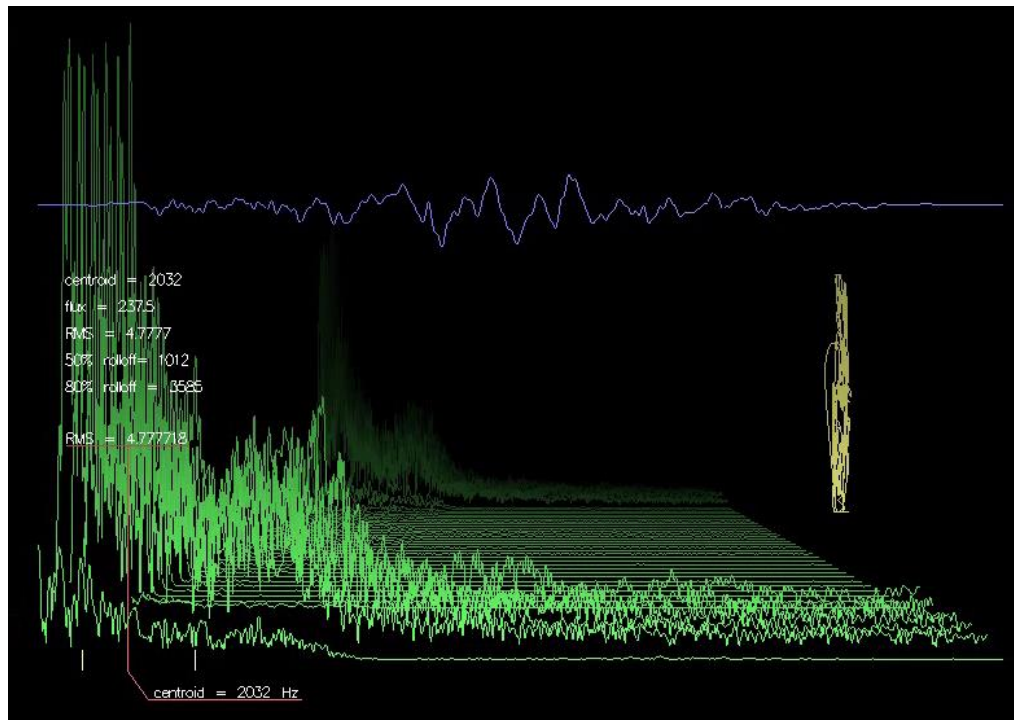
Pada Gambar 6.8 dapat dilihat ujicoba alat dengan jarak 3 meter. Pada grafik yang ditunjukkan terlihat perbedaan yang sedikit signifikan. Grafik biru merupakan gambaran frekuensi yang dihasilkan. Pada kurva biru terlihat bahwa jika nilai gain tidak terlalu besar maka lebar band frekuensi akan besar sehingga nilai Q kecil. Tetapi pada nilai gain yang besar maka lebar band frekuensi menjadi sempit. Oleh karena itu hasil respon keseluruhan dari equaliser grafik akan terbentuk gelombang panjang dan melebar. Pada grafik terlihat frekuensi berupa 1953 hz. Grafik hijau merupakan RMS yang dihasilkan berupa 3,9748. Grafik kuning adalah rolloff dan didapatkan roll off dalam rentang 926 sampai dengan 3122.



Gambar 6.8 Percobaan modul ESP8266 dengan selisih 3 meter

Pada Gambar 6.9 dapat dilihat ujicoba modul ESP8266 dengan selisih jangkauan 6 meter. Pada grafik terlihat Frekuensi yang mulai stabil. Grafik biru merupakan gambaran frekuensi yang dihasilkan. Pada kurva biru bahwa jika nilai gain tidak terlalu besar maka lebar band frekuensi akan besar sehingga nilai amplitudonya kecil. Tetapi pada nilai gain yang besar maka lebar band frekuensi menjadi sempit. Oleh karena itu hasil respon keseluruhan dari equaliser grafik akan terbentuk gelombang kecil-kecil yang melebar. pada grafik terlihat frekuensi berupa 2032 hz. Grafik hijau merupakan RMS yang dihasilkan berupa 4,777.

Grafik kuning adalah rolloff dan didapatkan roll off dalam rentang 1012 sampai dengan 3585.



Gambar 6.9 Percobaan modul ESP8266 dengan selisih 6 meter

6.3.2 Hasil dan Analisis

Hasil dan analisa didapatkan setelah melakukan uji coba alat, kemudian mencatat data yang didapatkan. Setelah melakukan ujicoba modul ESP8266, hasil yang didapat kemudian di jabarkan kedalam tabel 6.3.

Tabel 6.3 Hasil Pengujian pada Wifi

	Jarak		
Nama Timbral	1 meter	3 meter	6 meter
Centroid	2287 Hz	1963 Hz	2032 Hz
Flux	506,3	648,6	237,5
RMS	5.4770	3,964	4,777

Dari hasil pengujian pada Tabel 6.3 didapatkan analisis pengiriman suara yang dilakukan tidak terlalu mempengaruhi kualitas suara yang dihasilkan meskipun dengan jarak yang berbeda. Untuk kualitas suata sendin bisa dibilang mempunyai kualitas yang bagus dari kedua modul pengiriman lainnya.

6.4 Pengujian Keseluruhan

Dalam pengujian keseluruhan ketiga modul yang telah di uji coba dibandingkan secara bersamaan untuk menentukan hasil data.

Dari ketiga hasil analisa tersebut maka akan dirangkum kedalam tabel. Hasil pengujian dibagi menjadi tiga sesuai dengan jarak modul receiver dengan transmitter. Pada Ujicoba ketiga modul ini, akan dibandingkan data dari ketiga modul secara bersamaan.

Tabel 6.4 Perbandingan Pengujian modul dengan jarak 1 meter

	Modul pengiriman		
Nama Timbral	nrf	bluetooth	wifi
Centroid	3018 Hz	1791 Hz	2287 Hz
Flux	803,0	718,1	506,3
RMS	0.3500	0.1086	5.4770

Tabel 6.4 merupakan perbandingan pengujian ketiga modul dengan jarak 1 meter. Pada perbandingan dengan jarak 1 meter terlihat perbedaan centroid pada masing-masing modul. Modul NRF24L01 menghasilkan gelombang 3018 Hz, sedangkan untuk bluetooth menghasilkan gelombang suara sebesar 1791 Hz dan modul WIFI menghasilkan gelombang 2287 Hz. Semakin besar gelombang suara yang dihasilkan maka semakin nyaring suara bisa didengar.

Flux, yang biasa disebut *Sound Flux* atau *Acoustic Flux* merupakan tingkat di mana energi suara dipancarkan, dipantulkan, ditransmisikan atau diterima, per satuan waktu. Pada percobaan dengan jarak 1 meter modul NRF memiliki tingkat flux tertinggi, sedangkan modul bluetooth memiliki Flux dibawah tingkat modul NRF dan jaringan wifi memiliki tingkat Flux yang paling rendah.

RMS atau Root Mean Square adalah proses yang digunakan untuk menentukan output daya rata-rata speaker selama periode waktu yang digunakan. Semakin besar RMS maka semakin banyak daya yang digunakan oleh modul. Pada percobaan dengan jarak 1 meter modul NRF memiliki RMS 0,3500, modul Bluetooth memiliki RMS 0,1086, dan modul Wifi memiliki RMS tertinggi dengan nilai 5,4770.

Tabel 6.5 Perbandingan Pengujian modul dengan jarak 3 meter

	Modul pengiriman		
Nama Timbral	nrf	bluetooth	wifi
Centroid	1718 Hz	1120 Hz	1963 Hz
Flux	402,4	740,5	648,6
RMS	0,7304	0,7304	3,964

Tabel 6.5 merupakan perbandingan pengujian ketiga modul dengan jarak 3 meter. Pada perbandingan dengan jarak 1 meter terlihat perbedaan centroid pada masing-masing modul. Centroid pada Modul NRF24L01 turun menjadi 1718 Hz, sedangkan untuk bluetooth menghasilkan gelombang suara sebesar 1120 Hz dan modul WIFI menghasilkan gelombang 1963 Hz. Semakin besar gelombang suara yang dihasilkan maka semakin nyaring suara bisa didengar.

Pada percobaan dengan jarak 3 meter modul Bluetooth memiliki tingkat flux tertinggi, sedangkan modul Wifi memiliki Flux dibawah tingkat modul Bluetooth dan jaringan NRF memiliki tingkat Flux yang paling rendah.

Pada percobaan dengan jarak 3 meter modul NRF memiliki RMS 0,7304, modul Bluetooth memiliki RMS 0,7304, dan modul Wifi memiliki RMS tertinggi dengan nilai 3,964.

Tabel 6.6 Perbandingan Pengujian modul dengan jarak 6 meter

	Modul pengiriman		
Nama Timbral	nrf	bluetooth	wifi
Centroid	1090 Hz	1269 Hz	2032 Hz
Flux	179,8	691,3	237,5
RMS	2,9589	0,4518	4,777

Tabel 6.5 merupakan perbandingan pengujian ketiga modul dengan jarak 6 meter. Pada perbandingan dengan jarak 1 meter terlihat perbedaan centroid pada masing-masing modul. Modul NRF24L01 menghasilkan gelombang 1090 Hz, sedangkan untuk bluetooth menghasilkan gelombang suara sebesar 1269 Hz dan modul WIFI menghasilkan gelombang 2032 Hz. Semakin besar gelombang suara yang dihasilkan maka semakin nyaring suara bisa didengar.

Pada percobaan dengan jarak 6 meter modul Bluetooth memiliki tingkat flux tertinggi, sedangkan modul Wifi memiliki Flux dibawah tingkat modul Bluetooth dan jaringan NRF memiliki tingkat Flux yang paling rendah.

Pada percobaan dengan jarak 6 meter modul NRF memiliki RMS 2,9589, modul Bluetooth memiliki RMS 0,4518, dan modul Wifi memiliki RMS tertinggi dengan nilai 4,777.

Pada pengujian secara keseluruhan ini dilakukan dengan cara menggunakan ketiga modul dengan jarak yang sama. Pengujian ini akan menggunakan jarak 1 meter, 3 meter, dan 6 meter. Hasil pengujian menunjukkan perbandingan yang cukup besar saat menjalankan program secara bersamaan. Dari hasil pengujian ini menunjukkan pada nrf setiap sinyal suara yang dikirimkan mempunyai data yang berbeda secara signifikan. Setiap perbedaan jarak maka data yang didapat mengalami penurunan. Sedangkan pada bluetooth sinyal suara yang dihasilkan

relatif stabil. Tetapi terjadi penurunan kualitas suara. Untuk pengujian pada wifi hasil yang didapat pengiriman suara relatif stabil. Kualitas suara yang dihasilkan juga konsisten meskipun dilakukan pada jarak yang berbeda. Pada ujicoba yang telah dilakukan dapat disimpulkan setiap modul mempunyai kelebihan dan kekurangan masing-masing.

BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan dapat dijelaskan setelah melakukan pengujian sistem. Kemudian dirumuskan berdasarkan rumusan masalah yang ada.

1. Untuk mengimplementasikan pengiriman data suara secara wireless dan realtime dapat dilakukan dengan ketiga modul yang digunakan yaitu nrf, bluetooth, dan wifi. Semua proses yang ada dijalankan menggunakan mikrokontroler Arduino Uno.
2. Dalam melakukan pengujian didapatkan beberapa faktor yang terlibat dalam pengiriman suara secara wireless, diantaranya besarnya frekuensi suara, jarak antarmodule, dan noise yang dihasilkan. Faktor tersebut mempengaruhi kualitas suara yang dihasilkan. Semakin besar frekuensi yang dihasilkan maka kualitas suara semakin jelas. Sedangkan untuk jarak antar transmiter dan receiver berpengaruh pada beberapa modul.
3. Hasil pengujian menunjukkan perbandingan yang cukup besar saat harus mengeksekusi program secara bersama-sama. Hasil pengujian ini menunjukkan pada nrf setiap sinyal suara yang dikirimkan mempunyai data yang berbeda secara signifikan. Setiap perbedaan jarak maka data yang didapat mengalami penurunan. Sedangkan pada bluetooth sinyal suara yang dihasilkan relatif stabil. Tetapi terjadi penurunan kualitas suara. Untuk pengujian pada wifi hasil yang didapat pengiriman suara relatif stabil. Kualitas suara yang dihasilkan juga konsisten meskipun dilakukan pada jarak yang berbeda. Pengujian kepada beberapa modul yang dilakukan dapat disimpulkan module wifi mempunyai karakteristik yang ideal sebagai pengiriman suara.

7.2 Saran

Saran diberikan berdasarkan hasil penelitian ini dan dapat digunakan untuk penelitian selanjutnya.

1. Dapat menambahkan noise filter pada masing-masing module.
2. Pada penelitian berikutnya dapat membandingkan pengiriman suara dengan menggunakan module yang lebih baik.
3. Pada penelitian berikutnya diharapkan hasil penelitian bisa digunakan dalam kehidupan sehari-hari.

DAFTAR PUSTAKA

- Abdi Pramana., 2015. APA ITU WIRELESS SENSOR NETWORK ? Tersedia melalui <<https://sappame.wordpress.com/2015/10/10/wireless-sensor-network/>> [Diakses 24 Maret 2018]
- Arduino, 2016. *ARDUINO UNO*. [Online] Tersedia di <<https://arduino.cc/en/Main/ArduinoBoardUno>> [Diakses 25 Maret 2018]
- Bagus Seta Inba C, Henning Titi Ciptaningtyas, dan Hudan Studiawan,. 2014. RANCANG BANGUN APLIKASI KOMUNIKASI AUDIO PADA JARINGAN NIRKABEL LOKAL BERBASIS ANDROID MENGGUNAKAN ALGORITMA JOINT CODING RATE CONTROL, Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)
- Dargie, W. and Poellabauer, C. 2010. *Fundamentals of wireless sensor networks: theory and practice*. John Wiley and Sons. pp.
- Ecadio., 2017. Mengenal Arduino Uno R3. [Online] Tersedia melalui <<http://ecadio.com/mengenal-dan-belajar-arduino-uno-r3>> [Diakses 16 Maret 2018]
- ESP8266, 2014. ESP8266 [Online] Tersedia di <<http://esp8266.net/>> [Diakses 26 Maret 2018]
- Grant, A. E. & Meadows, J. H. 2010. *Communication Technology Update and Fundamentals*. 12th Edition. Focal Press
- HC-05 2010. HC-05 -Bluetooth to Serial Port Module. [Online] Tersedia di <<http://www.electronicaestudio.com/docs/istd016A.pdf>> [Diakses 25 Maret 2018]
- Isser H.Palendeng, Janny O. Wuwung, Ellia K. Allo, Benny S. Narasiang., 2012. Rancang Bangun Sistem Audio Nirkabel Menggunakan Gelombang Radio FM, Jurusan Teknik Elektro-FT, UNSRAT, Manado
- Lin-Huang Chang, Chao-Chieh Chen, Tsung-Han Lee,. 2010. *Voice Transmission over Wireless Sensor Networks*, Taichung, Taiwan.
- Nrf24l01, 2014. Tutorial Nrf24l01. [Online] Tersedia di <<https://courses.ideate.cmu.edu/16-223/f2014/tutorial-nrf24l01/>> [Diakses 25 Maret 2018]
- Suhaidah Binti Abu Bakar, 2009. *WIRELESS MICROPHONE SYSTEM*, Universiti Teknikal Malaysia Melaka
- Teori Elektronika, 2018. Sinyal Audio (Gelombang Suara) [Online] Tersedi di <<http://elektronika-dasar.web.id/sinyal-audio-gelombang-suara/>> [Diakses 18 Desember 2018]
- Tim Works,.2005. *Selection and Operation of Wireless Microphone Systems*, United States.

Prof. Deepali Shinkar , Amit Shinde , Puja Goswami , Sachin Shinde 2018. *Wireless Voice Transmission using WIFI and Bluetooth on Android Platform*. Dalam : *INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH AND ENGINEERING TRENDS*, February 2018

LAMPIRAN

A. Program Bluetooth

Program Program reciever bluetooth	
1	
2	<code>#include <AutoAnalogAudio.h></code>
3	<code>AutoAnalog aaAudio;</code>
4	
5	<code>void DACC_Handler(void)</code>
6	<code>{</code>
7	<code>aaAudio.dacHandler();</code>
8	<code>}</code>
9	
10	
11	<code>void setup()</code>
12	<code>{</code>
13	<code>pinMode(LED_BUILTIN, OUTPUT);</code>
14	<code>pinMode(A0, OUTPUT);</code>
15	<code>Serial.begin(57600);</code>
16	<code>Serial.println("Analog Audio Begin");</code>
17	<code>aaAudio.begin(0, 1); //Setup aaAudio using both DAC and ADC</code>
18	<code>aaAudio.autoAdjust = 1;</code>
19	<code>aaAudio.dacBitsPerSample = 16;</code>
20	<code>aaAudio.setSampleRate(26050);</code>
21	<code>}</code>
22	
23	<code>uint32_t dispTimer = 0;</code>
24	<code>uint8_t channelSelection = 0;</code>
25	
26	<code>void loop()</code>
27	<code>{</code>
28	<code> RX();</code>
29	<code>}</code>
30	
31	<code>uint32_t dynSampleRate = 0;</code>
32	
33	<code>void RX()</code>
34	<code>{</code>
35	<code> int dataNum = 0;</code>
36	<code> while (dataNum <= 15)</code>

37	{
38	while (Serial.available () && dataNum <= 15)
39	{
40	aaAudio.dacBuffer16[dataNum] = (uint8_t)Serial.read();
41	dataNum++;
42	}
43	}
44	aaAudio.feedDAC(channelSelection, 16);
45	}
46	

B. Program wifi

Program Wifi	
1	//USE ON ARDUINO 1.8.3
2	#include <Wire.h>
3	#include <ESP8266WiFi.h>
4	#include <WiFiClient.h>
5	#include <WiFiUdp.h>
6	#include "ESP8266mDNS.h"
7	#include <ArduinoOTA.h>
8	
9	const int mySDA = D7;
10	const int mySCL = D6;
11	const int AMPLI_MUTE_PIN = D1;
12	const int AMPLI_SHUTDOWN_PIN = D2;
13	
14	const int RIGHT_BTN = D3;
15	const int LEFT_BTN = D4;
16	const int LED1 = D8;
17	
18	const int udp_recv_port = 45990;
19	
20	WiFiUDP udp;
21	TwoWire i2c;
22	
23	#define NB_DATA_BUFS 5
24	uint16_t data_buf[NB_DATA_BUFS][700]; // data buffer, N buffered

```

25
26     unsigned int current_play_data_buf; // current data buf being played
27     unsigned int play_data_buf_pos; // position in the ADC data buffer
28     unsigned int current_recv_data_buf; // current data buf being received
29
30     bool play_waiting = true;
31     bool amplifier_stopped = false;
32     long play_waiting_at;
33
34     bool left_btn_pressed;
35     bool right_btn_pressed;
36
37     #define ICACHE_RAM_ATTR    __attribute__((section(".iram.text")))
38     #define twi_sda mySDA
39     #define twi_scl mySCL
40     #define twi_dcount 0
41     #define twi_clockStretchLimit 10
42     #define SDA_LOW()    (GPES = (1 << twi_sda)) //Enable SDA (becomes output
and since GPO is 0 for the pin, it will pull the line low)
43     #define SDA_HIGH()    (GPEC = (1 << twi_sda)) //Disable SDA (becomes input
and since it has pullup it will go high)
44     #define SDA_READ()    ((GPI & (1 << twi_sda)) != 0)
45     #define SCL_LOW()    (GPES = (1 << twi_scl))
46     #define SCL_HIGH()    (GPEC = (1 << twi_scl))
47     #define SCL_READ()    ((GPI & (1 << twi_scl)) != 0)
48
49
50     static void twi_delay(unsigned char v) {
51         unsigned int i;
52         #pragma GCC diagnostic push
53         #pragma GCC diagnostic ignored "-Wunused-but-set-variable"
54         unsigned int reg;
55         for (i = 0; i < v; i++) reg = GPI;
56         #pragma GCC diagnostic pop
57     }
58
59     static inline ICACHE_RAM_ATTR bool twi_write_start(void)
60     {
61         SCL_HIGH();
62         SDA_HIGH();
63         if (SDA_READ() == 0) return false;
64         SDA_LOW();
65         return true;

```

```

66     }
67
68     static inline ICACHE_RAM_ATTR bool twi_write_stop(void)
69     {
70         uint32_t i = 0;
71         SCL_LOW();
72         SDA_LOW();
73         SCL_HIGH();
74         while (SCL_READ() == 0 && (i++) < twi_clockStretchLimit); // Clock
75 stretching
76         SDA_HIGH();
77         return true;
78     }
79
80     static inline ICACHE_RAM_ATTR bool twi_write_bit(bool bit)
81     {
82         uint32_t i = 0;
83         SCL_LOW();
84         if (bit) SDA_HIGH();
85         else SDA_LOW();
86         twi_delay(twi_dcount + 1);
87         SCL_HIGH();
88         while (SCL_READ() == 0 && (i++) < twi_clockStretchLimit); // Clock
89 stretching
90         return true;
91     }
92
93     static inline ICACHE_RAM_ATTR bool twi_read_bit(void)
94     {
95         uint32_t i = 0;
96         SCL_LOW();
97         SDA_HIGH();
98         twi_delay(twi_dcount + 2);
99         SCL_HIGH();
100        while (SCL_READ() == 0 && (i++) < twi_clockStretchLimit); // Clock
101 stretching
102        bool bit = SDA_READ();
103        return bit;
104    }
105
106    static inline ICACHE_RAM_ATTR bool twi_write_byte(unsigned char byte)
107    {

```

```

108     unsigned char bit;
109     for (bit = 0; bit < 8; bit++)
110     {
111         twi_write_bit(byte & 0x80);
112         byte <<= 1;
113     }
114     return !twi_read_bit();//NACK/ACK
115 }
116
117 static inline ICACHE_RAM_ATTR unsigned char twi_read_byte(bool nack)
118 {
119     unsigned char byte = 0;
120     unsigned char bit;
121     for (bit = 0; bit < 8; bit++) byte = (byte << 1) | twi_read_bit();
122     twi_write_bit(nack);
123     return byte;
124 }
125
126
127 unsigned char inline ICACHE_RAM_ATTR mytwi_writeTo(unsigned char address,
128 unsigned char * buf, unsigned int len, unsigned char sendStop)
129 {
130     unsigned int i;
131     if (!twi_write_start()) return 4; //line busy
132     if (!twi_write_byte(((address << 1) | 0) & 0xFF))
133     {
134         if (sendStop) twi_write_stop();
135         return 2; //received NACK on transmit of address
136     }
137     for (i = 0; i < len; i++)
138     {
139         if (!twi_write_byte(buf[i]))
140         {
141             if (sendStop) twi_write_stop();
142             return 3; //received NACK on transmit of data
143         }
144     }
145     if (sendStop) twi_write_stop();
146     i = 0;
147     while (SDA_READ() == 0 && (i++) < 10)
148     {

```

```

149         SCL_LOW();
150         SCL_HIGH();
151     }
152     return 0;
153 }
154
155 static inline ICACHE_RAM_ATTR uint8_t DAC(uint16_t value)
156 {
157     /* value is 76543210 XXXXBA98
158        per the datasheet for fast write:
159        1 1 0 0 A2 A1 A0 0 <ACK> 0 0 PD1 PD0 D11 D10 D9 D8 <ACK> D7 D6 D5 D4
160        D3 D2 D1 D0 <ACK>
161     */
162
163     uint8_t buf[2] = { (value >> 8) & 0x0F, (value & 0xFF) };
164     int ret = mytwi_writeTo(0x60, buf, 2, true);
165     return ret;
166 }
167
168 void ICACHE_RAM_ATTR playsample_isr(void)
169 {
170     if (play_waiting)
171     {
172         return;
173     }
174     DAC(data_buf[current_play_data_buf][play_data_buf_pos]);
175     play_data_buf_pos++;
176
177     if (play_data_buf_pos >= sizeof(data_buf[0]) / sizeof(data_buf[0][0]))
178     {
179         play_data_buf_pos = 0;
180         current_play_data_buf++;
181
182         if (current_play_data_buf == NB_DATA_BUFS)
183         {
184             current_play_data_buf = 0;
185         }
186
187         if (current_play_data_buf == current_recv_data_buf)
188         {
189             play_waiting = true;

```

```

190         play_waiting_at = micros();
191     }
192 }
193 }
194
195 void ota_onstart(void)
196 {
197     // Disable timer when an OTA happens
198     timer1_detachInterrupt();
199     timer1_disable();
200 }
201
202 void ota_onprogress(unsigned int sz, unsigned int total)
203 {
204     Serial.print("OTA: ");
205     Serial.print(sz);
206     Serial.print("/");
207     Serial.print(total);
208     Serial.print("=");
209     Serial.print(100 * sz / total);
210     Serial.println("%");
211 }
212
213 void ota_onerror(ota_error_t err)
214 {
215     Serial.print("OTA ERROR:");
216     Serial.println((int)err);
217 }
218
219 void left_btn_intr(){
220     left_btn_pressed = 1;
221 }
222
223 void right_btn_intr(){
224     right_btn_pressed = 1;
225 }
226
227 void setup ( void ){
228     Serial.begin ( 115200 );
229
230     // Serial.println("I was built on " __DATE__ " at " __TIME__ "");

```

```

231     i2c.begin(mySDA, mySCL);
232     i2c.setClock(400000);
233
234     WiFi.mode(WIFI_STA);
235     // WiFi.begin("freewifi", "12345678");
236     WiFi.begin("Redmi3X", "aciedsavior");
237     WiFi.setSleepMode(WIFI_MODEM_SLEEP);
238
239     Serial.println("Connecting to wifi");
240     while ( WiFi.status() != WL_CONNECTED )
241     {
242         delay ( 500 );
243         Serial.print ( "." );
244     }
245
246     Serial.println ( "" );
247     Serial.print ( "Cnnected to " );
248     Serial.println ( "g1" );
249     Serial.print ( "IP " );
250     Serial.println ( WiFi.localIP() );
251
252     ArduinoOTA.onStart(ota_onstart);
253     ArduinoOTA.onError(ota_onerror);
254     ArduinoOTA.onProgress(ota_onprogress);
255     ArduinoOTA.setHostname("bb-recv");
256     ArduinoOTA.begin();
257
258     timer1_isr_init();
259     timer1_attachInterrupt(playsample_isr);
260     timer1_enable(TIM_DIV16, TIM_EDGE, TIM_LOOP);
261     timer1_write(clockCyclesPerMicrosecond() / 16 * 50); //50us = 20 kHz
262     sampling freq
263
264     udp.begin(udp_recv_port);
265
266     pinMode(AMPLI_MUTE_PIN, OUTPUT);
267     pinMode(AMPLI_SHUTDOWN_PIN, OUTPUT);
268     digitalWrite(AMPLI_SHUTDOWN_PIN, 0);
269     digitalWrite(AMPLI_MUTE_PIN, 0);
270
271     pinMode(LEFT_BTN, INPUT_PULLUP);
272     attachInterrupt (digitalPinToInterrupt (LEFT_BTN),          left_btn_intr,

```



```

272 FALLING);
273     pinMode(RIGHT_BTN, INPUT_PULLUP);
274     attachInterrupt(digitalPinToInterrupt(RIGHT_BTN),      right_btn_intr,
275 FALLING);
276
277     pinMode(LED1, OUTPUT);
278     digitalWrite(LED1, 0);
279
280 }
281
282 int do_undelta7(const uint8_t *val, int sz, uint16_t *out)
283 {
284     // Implement delta 7 decompression.
285     // First bit = 0 <=> uncompressed 15 bits following
286     // First bit = 1 <=> 7 bits follow representing delta
287     // must switch to big endian...
288     uint16_t last = 0;
289     uint8_t *ptr = (uint8_t *)&out[0];
290     const uint8_t *start = ptr;
291     for (int i = 0; i < sz; i++)
292     {
293         uint16_t *ptr16 = (uint16_t *)ptr;
294         const int8_t firstbyte = val[i];
295         if (firstbyte & 0x80)
296         {
297             // Delta7 compressed
298             // byte is CSMMMMMM
299             int8_t delta = firstbyte & 0x3F;
300             if (firstbyte & 0x40)
301             {
302                 delta = -delta;
303             }
304             const uint16_t value = last + delta;
305             *ptr16 = value;
306             ptr += 2;
307             last = value;
308         }
309         else
310         {
311             // uncompressed -- switch bytes back to LE
312             *ptr++ = val[i + 1];

```

```

313         *ptr++ = val[i];
314         last = val[i + 1] | val[i] << 8;
315         i++;
316     }
317 }
318 return ptr - start;
319 }
320 void loop ( void )
321 {
322     ArduinoOTA.handle();
323     int sz = udp.parsePacket();
324     if (sz)
325     {
326         uint8_t buf[sz];
327         udp.read(&buf[0], sz);
328         current_rcv_data_buf++;
329         // Serial.println(sz);
330         Serial.println(buf[sz]);
331         if (current_rcv_data_buf == NB_DATA_BUFS)
332         {
333             current_rcv_data_buf = 0;
334             if (current_rcv_data_buf == current_play_data_buf &&
335 !play_waiting)
336             {
337                 Serial.println("buffer overflow when receiving");
338             }
339         }
340         do_undelta7(buf, sz, &data_buf[current_rcv_data_buf][0]);
341         if (play_waiting)
342         {
343             Serial.print("Restarting play, was waiting (us)");
344             Serial.println(micros() - play_waiting_at);
345             // Re-enable *then* unmute in that order to avoid pops
346             digitalWrite(AMPLI_SHUTDOWN_PIN, 1);
347             digitalWrite(AMPLI_MUTE_PIN, 1);
348             play_waiting = false;
349             amplifier_stopped = false;
350             digitalWrite(LED1, 1);
351         }
352         Serial.println("");
353     }

```

```

354         // If not playing anything, but amplifier is still up
355         if (!amplifier_stopped && play_waiting)
356         {
357             if ((micros() - play_waiting_at) > 2000 * 1000)
358             {
359                 // If nothing has been played for two seconds, shut down the
360                 amplifier
361                 Serial.println("Shutting down amplifier!");
362                 digitalWrite(AMPLI_SHUTDOWN_PIN, 0);
363                 digitalWrite(AMPLI_MUTE_PIN, 0);
364                 amplifier_stopped = true;
365                 digitalWrite(LED1, 0);
366             }
367         }
368         if (left_btn_pressed)
369         {
370             left_btn_pressed = 0;
371             digitalWrite(AMPLI_MUTE_PIN, 0);
372             digitalWrite(AMPLI_SHUTDOWN_PIN, 0);
373             Serial.println("tombol_kiri");
374         }
375         if (right_btn_pressed)
376         {
377             digitalWrite(AMPLI_SHUTDOWN_PIN, 1);
378             digitalWrite(AMPLI_MUTE_PIN, 1);
379             Serial.println("tombol_kanan");
380             udp.beginPacket(udp.remoteIP(), 45990);
381             udp.write("sendnow");
382             udp.endPacket();
383             right_btn_pressed = 0;
384         }
385         // If the amplifier is stopped, add a delay for power saving
386         if (amplifier_stopped)
387         {
388             delay(10);
389         }
390         // Serial.println(buf);
391     }

```

